Kristiania

Towards Universal Transport Mode Detection

A Design Science Approach for Unifying Data Frameworks and Platforms

Anders Skretting

PhD Dissertations Kristiania 2025:3

Towards Universal Transport Mode Detection

A Design Science Approach for Unifying Data Frameworks and Platforms

Anders Skretting

Kristiania

Thesis submitted for the degree of Philosophiae Doctor (PhD) in
Applied Information Technology
Kristiania University of Applied Sciences

Fall 2025

PhD Dissertations Kristiania no.3, 2025

School of Doctoral Studies

Kristiania University of Applied Sciences

Oslo, Norway

ISSN: 3084-1488 (print)

ISSN: 3084-147X (online)

ISBN: 978-82-93953-11-1 (print)

ISBN: 978-82-93953-12-8 (online)

Printed at Byråservice

Kristiania

Acknowledgements

Completing this PhD has been a journey of intellectual growth, perseverance, and personal sacrifice, and I could not have done it without the support and guidance of many individuals and institutions.

First and foremost, I am deeply grateful to my main supervisor, Tor-Morten Grønli, for his invaluable guidance, expertise, and support throughout this process. Your insightful feedback, patience, and encouragement have been instrumental in shaping both this research and my development as a researcher. I am also sincerely thankful to my co-supervisor, Raghava Rao Mukkamala, for his thoughtful advice, constructive criticism, and continued support, which have significantly contributed to this work. As the majority of this research has been conducted in my spare time, I am especially grateful to both of you for the flexibility and understanding you have shown me throughout this process.

I would like to express my gratitude to my current employer, Kripos, for their flexibility and understanding in allowing me, on occasion, to dedicate time to my research during quieter periods. Their support, has been invaluable in helping me balance my professional responsibilities with my academic pursuits.

A special thank you goes to my fellow PhD student, Mathilde Hogsnes, whose support, discussions, and camaraderie have made this challenging journey far more manageable. Your encouragement and shared experiences have been a source of motivation and reassurance. Without our regular rants at the bar, I doubt I would have endured this period with my sanity intact.

I am also grateful to my previous employer, Kogenta, for initiating this project and for their continued interest in its development. Their initial vision and ongoing engagement have played a key role in shaping the research and its real-life implications.

Lastly, I owe an immense debt of gratitude to my partner, Siv Kathrine Jin Skorstad, for her support, patience, and understanding during this demanding period. Your encouragement and belief in me have been my anchor through the highs and lows of this journey, and I could not have done this without you. Quite literally, without your cooking, I might not have made it at all as I would have succumbed to sheer lack of sustenance.

Abstract in English

The rapid urbanisation of modern societies has led to an increased demand for efficient, sustainable, and intelligent transportation solutions. Intelligent Transportation Systems play a crucial role in addressing these challenges by leveraging emerging technologies to optimise mobility services and enhance public transport accessibility. A key component of Intelligent Transportation Systems is transport mode detection, which utilises smartphone sensor data and machine learning to infer how individuals travel. This capability enhances mobility analytics, enables real-time travel behaviour analysis, and supports automated fare collection systems, reducing reliance on manual ticketing while improving public transport efficiency. However, existing transport mode detection methods often depend on centralised processing or external infrastructure, leading to challenges such as latency, increased operational costs, and privacy concerns. Additionally, a lack of standardised methodologies and datasets results in significant discrepancies between existing approaches, complicating comparisons and limiting the generalisability of research findings. Following a Design Science research methodology, this thesis explores various facets of transport mode detection through an iterative process of model development, evaluation, and refinement, with a particular emphasis on on-device solutions for mobile devices. A key contribution of this thesis is the creation of a diverse and representative dataset, spanning multiple devices, operating systems, and transport environments, which facilitates the development of methods that generalise effectively across real-life conditions. Furthermore, a structured framework for feature importance and reduction is introduced, systematically identifying the most relevant features to enhance classification performance while minimising computational overhead, making models more suitable for resource-constrained devices. Additionally, various machine learning techniques, including deep learning and traditional classifiers, are employed and evaluated. The results demonstrate the ability to infer transport modes across a wide range of modalities and operating systems, contributing toward more practical real-life implementations. Through multiple iterations, this research develops and evaluates a lightweight, platform-agnostic framework for transport mode detection, demonstrating its practical applicability in real-life scenarios with minimal computational overhead. By contributing to the standardisation of methodologies

through the creation of a diverse dataset, a structured feature-importance framework, and a platform-agnostic framework that ensures cross-platform compatibility and reduces computational overhead for mobile devices, this thesis advances the fields of Intelligent Transportation Systems and Smart Mobility. The findings provide a foundation for future innovations in data-driven mobility services, supporting the transition toward adaptive, inclusive, and more efficient public transportation networks.

Abstract in Norwegian

Den raske urbaniseringen av samfunnet har ført til økt etterspørsel etter effektive, bærekraftige og intelligente transportløsninger. Forskningsfeltet "intelligente transportsystemer" spiller en avgjørende rolle i å møte disse utfordringene ved å utnytte nye teknologier for å optimalisere mobilitetstjenester og forbedre tilgjengeligheten til kollektivtransport. En sentral komponent i intelligente transportsystemer er transportmodusdeteksjon, som bruker sensordata fra smarttelefoner og maskinlæring for å fastslå hvordan enkeltpersoner reiser. Denne funksjonaliteten forbedrer mobilitetsanalyse, muliggjør sanntidsanalyse av reiseatferd og støtter automatiserte billettsystemer, noe som reduserer avhengigheten til manuell billettering samtidig som effektiviteten i kollektivtransporten økes. Eksisterende metoder for transportmodusdeteksjon er imidlertid ofte avhengige av sentralisert prosessering eller ekstern infrastruktur, noe som medfører utfordringer knyttet til dataforsinkelser, økte driftskostnader og personvern. I tillegg fører mangel på standardiserte metoder og datasett til betydelige forskjeller mellom eksisterende tilnærminger, noe som kompliserer sammenligninger og begrenser generaliserbarheten av forskningsresultater. Ved å følge en Design Science forskningsmetodologi utforsker denne avhandlingen ulike aspekter ved transportmodusdeteksjon gjennom en iterativ prosess med modellutvikling, evaluering og forbedring, med særlig vekt på lokale løsninger for mobile enheter. Et sentralt bidrag i dette arbeidet er utviklingen av et variert og representativt datasett som dekker flere enheter, operativsystemer og transportmoduser, og som muliggjør utviklingen av metoder som generaliserer effektivt under reelle forhold. Videre introduseres et strukturert rammeverk for å evaluere og redusere antall inputvariabler, gjennom å systematisk identifisere de mest relevante. Dette gjør det mulig å forbedre klassifikasjonsytelsen samtidig som beregningskostnaden reduseres, noe som gjør modellene mer egnet for enheter med begrensede ressurser. I tillegg er det anvendt og evaluert ulike maskinlæringsteknikker, inkludert dyp læring og tradisjonelle klassifikasjonsmetoder. Resultatene demonstrerer evnen til å identifisere transportmoduser på tvers av ulike modaliteter og operativsystemer, og dermed bidrar til mer praktiske implementeringer i reelle scenarier. Gjennom flere iterasjoner utvikler og evaluerer denne forskningen et lettvekts, plattformagnostisk rammeverk for transportmodusdeteksjon, og demonstrerer dets praktiske anvendelighet i

virkelige scenarier med minimal beregningskostnad. Ved å bidra til standardisering av metoder gjennom utviklingen av et variert datasett, et strukturert rammeverk for evaluering av input-variabler, samt et plattformagnostisk rammeverk som sikrer plattformkompatibilitet og redusert beregningskostnad for mobile enheter, fremmer dette arbeidet feltene ”intelligente transportsystemer” og ”smart mobilitet”. Funnene gir et solid grunnlag for fremtidige innovasjoner innen datadrevne mobilitetstjenester og støtter overgangen til mer adaptive, inkluderende og effektive kollektivtransportsystemer.

Contents

List of Articles

Articles Included in This Thesis:

1. **Survey of Automated Fare Collection Solutions in Public Transportation**
   Published in Journal of IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 9, pp. 14248–14266, 2022.

2. **Neural Network for Public Transport Mode Inference on Mobile Devices**
   Published in 2022 Springer 18th International Conference, Mobile Web and Intelligent Information Systems, pp. 65–78, 2022.

3. **Enhanced Transport Mode Recognition on Mobile Devices**
   Published in 2024 Proceedings of the 57th Hawaii International Conference on System Sciences, 2024.

4. **Feature Importance from Novel Dataset for On-Device Transport Mode Detection**
   Published in Journal of Taylor & Francis Intelligent Transportation Systems, 2025.

5. **Implementation and Evaluation of Cross-Platform, Lightweight, On-Device Transport Mode Detection**
   In review at Journal of IEEE Transactions on Intelligent Transportation Systems.

Articles Not Included in This Thesis:

- **Baseline for Performance Prediction of Android Applications**
  Published in 2020 IEEE International Conference on Big Data (Big Data), pp. 3304-3310, 2021.
- **Distributed Sensor Data Collection Using Mobile Clouds for Public Transportation**
  Published in 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 61–68, 2021.

List of Attachments

1. **Co-Author Declarations**

Part I

Thesis

Chapter 1

Introduction

More than half of the global population currently resides in urban areas, a proportion projected to increase to 68% by 2050 [1]. This rapid urban expansion presents significant challenges, particularly in transportation infrastructure. As cities grow, increased travel demand leads to congestion, environmental degradation, and inefficiencies in mobility [2]–[4]. These issues have driven research into innovative solutions that enhance the sustainability and efficiency of urban environments, one of which is the concept of smart cities [5]. Smart cities leverage technology to optimise urban processes, improve decision-making, and enhance sustainability [3], [5]–[8]. A core component of this vision is Intelligent Transportation Systems (ITS) [3], which focus on improving mobility through real-time data analysis and automation. Rather than addressing urbanisation broadly, ITS specifically enhances transportation networks, aiding citizens, companies, and governments in optimising mobility solutions [6]. ITS encompasses a range of applications, including solutions for public and private transport, with an emphasis on efficiency and safety across transportation systems [9].

Public transportation plays a critical role in addressing urban mobility challenges by reducing emissions and alleviating congestion in densely populated areas [10], [11]. The integration of smart technologies further enhances its potential. By utilising technological advancements, such as sensor and network technology, substantial amounts of data regarding vehicles and travellers can be gathered. This data forms the basis for real-time insights into issues such as traffic congestion, delays, and travel behaviour [12], [13]. The widespread adoption of smartphones has further improved these capabilities, enabling seamless data collection and real-time optimisation of public transportation services. Leveraging advancements in sensor technology, network connectivity, and smartphone-enabled data collection, machine learning technologies can be applied to the collected data to analyse and predict trends, whether past, present, or future. This capability allows policy makers, operators, and other stakeholders to make informed decisions when planning infrastructure, such as new public transportation routes and road construction [6], [12]. Real-time, automatic predictions of travellers' modes of transportation can also enable fully autonomous ticketing solutions, enhancing the efficiency, inclusivity, and overall user experience of public transportation services [14].

The primary focus of this thesis is the integration of mobile technologies and machine learning to develop efficient, on-device, platform-agnostic transport mode detection solutions. Specifically, it seeks to address key challenges related to predictive accuracy, platform compatibility, feature importance, and computational efficiency in transport mode detection. To this end, this thesis investigates machine learning models and generalisable feature importance techniques to construct frameworks that are evaluated in real-world contexts on mobile devices, ensuring both platform compatibility and computational efficiency.

1.1 Intelligent Transportation Systems for Smart Mobility

The foundational ideas behind Intelligent Transportation Systems (ITS) can be traced back to the 1930s [15], however the modern ITS framework began taking shape in the 1980s, driven by a core group of transportation professionals who recognised the transformative potential of emerging computing and communication technologies on highway transportation [9]. From these origins, ITS has grown substantially, expanding far beyond its initial highway-focused applications. Today, ITS is a broad, interdisciplinary field, encompassing areas such as transportation management, infrastructure control, operations, and policy-making [16]. Its applications now extend across various areas of transportation, including navigation services, railway, maritime, and aviation systems [17]. Notably, significant advancements in ITS have emerged from regions like Europe, the United States, and Japan, with more recent contributions from South Korea and Singapore [15], [18].

The evolution of ITS has paralleled advances in computing technology, shifting from early, infrastructure-based, one-way communication models, to complex multi-modal systems. Modern ITS solutions now integrate smartphones, vehicles, infrastructure, and contextual data to provide comprehensive mobility solutions [16]. Modern ITS systems are increasingly data-driven, leveraging data analytics and machine learning to address contemporary transportation challenges [12], [19], [20]. ITS is generally structured around six core components: Advanced Transportation Management Systems (ATMS), Advanced Traveller

Information Systems (ATIS), Advanced Vehicle Control Systems (AVCS), Commercial Vehicle Operations (CVO), Advanced Public Transportation Systems (APTS), and Advanced Rural Transportation Systems (ARTS) [12], [15], [20]. This research situates itself within the APTS category, which focuses on technologies that enhance the efficiency and operational quality of high-occupancy transport modes, such as buses and trains. Broadly, ITS aims to enhance the efficiency, safety and convenience of transportation systems for both people and goods [9]. While ITS has traditionally emphasised large-scale, infrastructure-focused systems, a broader and more holistic vision known as Smart Mobility has emerged.

Smart Mobility represents a holistic and evolving vision, with research in this field still in its infancy. Consequently, no standard definitions have yet been established [21]. However, Smart Mobility extends the infrastructure-oriented approach of ITS by emphasising multimodal integration, reducing environmental impact, and enhancing user experiences [22], [23]. It includes various environmentally focused initiatives, such as reducing private vehicle use and integrating transport modes to decrease emissions [24]. Generally, Smart Mobility employs digital technologies to integrate systems and transport modes, interacting with users to foster a sustainable, safe, and accessible environment that meets citizens' mobility needs [22].

Aligned with global sustainability goals [23], Smart Mobility seeks to reduce urban congestion, enhance environmental sustainability, and create integrated travel experiences across diverse transportation modes [22], [24]. It consists of actions designed to facilitate user mobility whether by foot, bicycle, or public and private transport, with a shared objective of reducing economic, environmental, and time costs [24]. A key aspect of Smart Mobility is public access to real-time information, which enhances service efficiency by saving time, improving the travel experience, lowering costs, and reducing CO_2 emissions [24]. Smart Mobility systems collect data from multiple sources, including traffic management systems, transport schedules, citizen-provided crowd data, and sensor inputs from vehicles, traffic lights, parking areas, and roads [21]. This multidisciplinary field intersects with diverse technologies and has evolved from the convergence of digital advances with the transportation sector, resulting in new methods for enhancing transportation network efficiency [22]. Solutions within Smart Mobility often rely on technologies such as machine learning in combination with large amounts of data gathered from interconnected devices [23]. As a result, this research is positioned not only within the domain of ITS but also contributes to Smart Mobility by enhancing user experience, inclusivity, and convenience in public transportation through enabling seamless automated fare collection, ultimately supporting sustainable and user-centred public transportation solutions.

1.2 Automated Fare Collection and Be-In/Be-out

Automated fare collection (AFC) in public transportation are integral components of Smart Mobility initiatives [24] and Intelligent Transportation Systems (ITS) [15]. AFC systems have already been widely implemented in urban transit networks [20], [25] and encompasses the technologies and processes used to detect passenger boarding events and to process fare payments [26]. Typically, current AFC implementations rely on the use of tokens or smart cards to identify check-in and check-out activities for fare calculation purposes [26]. Prominent examples of AFC solutions in Europe include the *Andante Card* in Portugal, the *OV chipcard* system in the Netherlands, and the contactless payment system, *Oyster*, in London, UK. In AFC solutions, passengers utilise smart cards or similar token-based devices to register entry and exit from public transportation vehicles. Such AFC systems significantly facilitate other ITS applications, as the data they generate serves as a key resource for analysing passenger movement patterns [20]. However, current AFC systems face a number of notable challenges. Among these challenges are prolonged boarding times caused by manual check-in processes, which frequently require the use of physical payment mediums such as cards or tokens [26]. This requirement necessitates that passengers carry physical items, which can be inconvenient. This inconvenience is particularly pronounced for individuals with cognitive functional limitations, for whom managing physical payment methods can be a considerable source of stress [27].

As a result, the concept of Be-In/Be-Out (BIBO) have emerged. BIBO enhances travel convenience by automatically detecting when passengers enter or exit transportation services, simplifying the payment process and minimising manual interactions [28]–[32]. BIBO solutions differ from traditional public transport ticketing paradigms, where passengers are required to explicitly check in or check out using either physical or digital tickets. In contrast, BIBO systems aim to remove the need for active interaction from passengers by detecting their presence passively [29]–[35]. The concept of BIBO can improve existing automated fare collection (AFC) by allowing tickets to be automatically issued based on the user's confirmed presence on a transport service, thereby eliminating the need for manual ticketing processes [28]–[31], [33], [34]. The primary challenge in implementing BIBO in public transportation lies in achieving accurate in-vehicle presence detection mechanisms [28], [36], [37]. For fare collection to occur without human intervention, it is crucial that the system reliably determines a user's presence, preventing the deduction of funds from individuals who are not actually utilising the service. Several methods exist for establishing user presence within public transport vehicles.

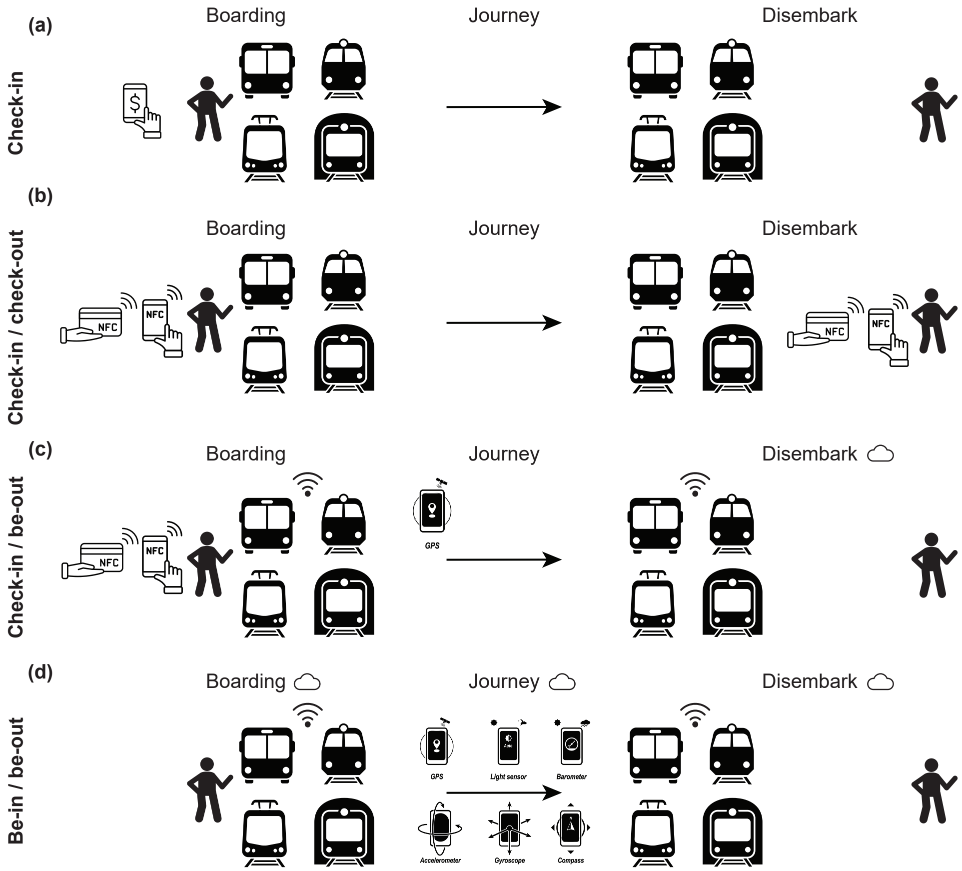


Figure 1.1: **Approaches to mobile ticketing in public transportation.** **(a) Check-in (CI):** the user is left to select and purchase the most suitable ticket prior to departure based on available fare information and leaves the vehicle upon arrival; **(b) Check-in/check-out (CICO):** the user actively checks in using a mobile application or digital reader before boarding and checks out upon arrival to make use of optimised fares; **(c) Check-in/be-out (CIBO):** the user actively checks in before boarding using a mobile application or reader, location data are acquired and communicated between the mobile device and a back-end server to offer optimised fare upon disembarkation; **(d) Be-in/be-out (BIBO):** active check-in and check-out are omitted, mobile sensor data are acquired and communicated to the back-end server during the journey to recognise the transportation mode and destination to offer optimised fares. [38]

The most common solutions involve some form of check-in and check-out mechanism, such as scanning a QR code, reading an RFID tag, or manually checking in via a software application [32], [39]. However, manual methods are not in line with the BIBO concept since they require the traveller to actively interact with the system, which can be time-consuming, cause boarding and disembarking delays, and ultimately reduce the efficiency of the public transport service. Additionally, manual methods are not fully inclusive, as explicit interactions may be challenging for elderly passengers or individuals with physical or cognitive disabilities [27]. The optimal solution would eliminate the need for explicit user interaction, embracing the BIBO concept, where user presence is detected passively.

BIBO solutions can be implemented through various approaches, using different sensors and technologies to address the challenge of passively establishing the presence of travellers. One common method involves installing hardware within public transport vehicles to detect passengers, often utilising technologies such as Bluetooth low-energy (BLE) [31], [33], [36], [40], [41], which has been identified as a key enabler in achieving BIBO solutions [36]. However, this approach faces limitations, such as difficulty distinguishing individual passengers and the requirement for BLE to remain active throughout the trip [28], [33]. Furthermore, passengers located outside the vehicle but within the BLE signal range may inadvertently be subject to ticket issuance [34]. In order to address this challenge researchers are now applying novel machine learning techniques in combination with smartphone sensor data to increase the precision of in-vehicle presence detection in order to enable accurate automated fare collection solutions [14], [37], [42].

1.3 Smartphone Sensor Data and Transport Mode Detection

In-vehicle presence detection systems rely on contextual information [43]. Smartphones are particularly well-suited to provide such contextual information due to their widespread availability and integrated sensors [44], [45]. As such, data from mobile devices serve as the largest source of information on human mobility [46]. Modern smartphones are equipped with numerous sensors capable of generating substantial amounts of data. However, not every sensor is relevant for in-vehicle presence detection. The relevant smartphone sensors can be categorised into three primary classes: inertial sensors, ambient sensors, and location sensors. Inertial sensors measure the physical motion and orientation of the device, while ambient sensors capture environmental conditions such as light or temperature levels. Location sensors on the other hand, such as the Global Positioning System (GPS), determine the geographical position of the device. By using data from inertial, ambient and location sensors in combination with machine learning, it is possible to detect the presence of a user or to infer the user's mode of transportation [37], [47]. The most widely used mobile operating systems, Android and iOS, offer a variety of sensors, including accelerometers, gyroscopes, and magnetometers [48]–[50]. The various raw sensor data can also be combined, in a process called *sensor fusion* [51], to form new sensors available through the smartphone operating systems, such as rotation vectors or gravity sensors. Data from smartphone sensors can be utilised in order to develop solutions for intelligent public transportation [48]–[50]. Examples of such systems are in-vehicle presence detection and transport mode detection (TMD).

TMD solutions can be categorised into three main approaches: statistical methods, rule-based methods, and machine learning methods [52]. Statistical methods and rule-based approaches typically struggle to differentiate between similar transport modes with high accuracy, such as cars, buses, and trains [52], [53]. In contrast, machine learning approaches offer better generalisation and improved discrimination between similar modes of transportation [54]. Therefore, this thesis focuses on employing machine learning methods towards improving critical aspects relating to local transport mode detection.

Machine learning algorithms are integral to processing smartphone sensor data for TMD and in-vehicle presence detection [55], [56]. While in-vehicle presence detection focuses on identifying whether a user is inside a specific vehicle, TMD aims to classify the overall mode of transportation the traveller is using (e.g., walking, cycling, or taking a bus). TMD can be used to effectively manage the challenge of detecting whether passengers are inside a vehicle rather than outside through accurate classification of the transportation mode. Thus, both challenges are closely related and can be addressed using data from smartphone sensors [37], [55], [57].

Furthermore, TMD is also closely related to human activity recognition (HAR), which applies similar techniques to infer a user's activity rather than the mode of transport they are using. HAR is central to mobile computing, as it automates the recognition of a user's activities [58]. Traditionally, HAR has been associated with computer vision research, however, with the increasing prevalence of mobile devices and wearables, the focus has shifted towards leveraging inertial sensors (e.g., accelerometers and gyroscopes) instead of images or videos [59]. Consequently, TMD can be considered a subset of HAR, and many techniques developed for HAR can be adapted to TMD [60], [61]. In fact, most research into TMD adopt approaches similar to HAR, where raw sensor data is processed to extract features, which are then used to train machine learning models for classification [62]. This growing emphasis on sensor-based techniques reflects a wider shift toward leveraging ubiquitous smartphone data combined with machine learning to better understand and classify human activities and mobility.

Although this shift towards employing smartphone sensor data in combination with machine learning is notable within the field of ITS and Smart Mobility, several challenges still remain. Predominantly, existing systems for in-vehicle presence detection and transport mode detection (TMD) depend on in-vehicle equipment and centralised architectures [47], [55], [57], [63], which can lead to elevated infrastructure costs [14], [26]. Today, mobile devices represent the largest source of data on human transportation patterns [46] and the constant data transmission between travellers' devices and onboard equipment or centralised servers can result in latency [64] and potential costs to users.

Centralised systems also present significant privacy concerns, primarily due to the continuous transfer of data from user devices to operators [65]. The transmitted data, especially precise location information, can be cross-referenced with external datasets to identify individuals or reveal sensitive details, such as home and work addresses, personal movement patterns, and affiliations inferred from visits to specific locations [66]. While data from inertial sensors is less precise, it can still pose privacy risks by enabling inferences about a user's location, identity, demographics, activities, and other characteristics when combined with additional data sources [67]. These privacy concerns have driven a growing interest in solutions that operate locally on users' devices, avoiding centralised data processing [56], [65], [68].

Another challenge is that existing solutions demonstrate varying levels of accuracy, particularly when applied across a diverse range of transportation modes. Furthermore, only a limited number of studies have been implemented and evaluated on real-life devices, where performance can differ significantly compared to controlled settings [56]. While a variety of studies propose solutions for TMD [69]–[75], the proposed solutions often differ significantly in the range of transport modes they can classify and the data being used to train the models. Consequently, there are no established best practices regarding the selection of sensors, algorithms, or similar methodologies that yield optimal performance in the context of TMD. Moreover, nearly all existing studies propose solutions specifically for Android devices. Although some address iOS [76]–[79], very few works address cross-platform capabilities of local TMD on mobile devices [56]. These challenges may stem from the limited availability of comprehensive datasets. Existing TMD datasets are scarce, and those that are available often exhibit significant limitations, such as insufficient representation of participants, devices, operating systems, geographical regions, and transportation modes. This lack of standardised benchmarking datasets further hampers the systematic evaluation and comparison of solutions [80].

1.4 Research Questions and Contributions

In order to embrace the ongoing shift from centralised to decentralised mobile solutions, and to support more privacy-friendly and cost-effective approaches, it becomes increasingly important to investigate various aspects of transport mode detection from a mobile solutions perspective. Given that mobile devices are inherently resource-constrained, a crucial consideration is computational efficiency, which itself depends on several factors. One such factor relates to the type of data used for inference, as different data sources provide varying insights and impose distinct computational requirements that may influence the overall efficiency of the solution. The significance lies not only in the nature of the data, but also in the number of data sources employed, as an increased variety of inputs may lead to greater model complexity. This can result in longer inference times and higher energy consumption, potentially diminishing the user experience. It is therefore essential to examine which data sources are most critical for accurately inferring the mode of transportation and which could potentially be excluded in order to optimise computational efficiency. Such investigations necessitate access to large, diverse and representative datasets that encompass a wide range of individuals, devices and data types. While researchers have begun to explore certain aspects of on-device transport mode detection using mobile data, much of the existing work remains platform dependent rather than platform agnostic, which constitutes a central focus of this thesis. Although there are multiple dimensions to on-device transport mode detection, this thesis is guided by the following overarching research question:

RQ0: *How can efficient on-device, platform-agnostic transport mode detection be achieved on mobile devices?*

In order to provide an answer to this overarching research question, different supporting research questions have been formulated, with different emphasis on accuracy, features and real-life evaluations:

RQ1: *How can machine learning models achieve high accuracy in transport mode detection across diverse transport modes, ensuring generalisability in real-life applications?*

RQ2: *How can a standardised framework for feature evaluation and reduction systematically identify relevant features, ensuring consistency and enabling reliable feature reduction across machine learning models?*

RQ3: *How can transport mode detection models be optimised for real-time, low-latency inference while maintaining computational efficiency on mobile devices?*

In the context of this thesis, generalisability refers to the ability of the developed artefact to effectively generalise across hardware, software, and users. To ensure that the findings of this thesis have broad implications, the primary focus of investigation has been smartphones. Consequently, within the scope of this thesis, mobile devices should be understood as referring to smartphones. However, the findings extend beyond smartphones and are also applicable to other mobile and resource-constrained devices. These considerations provide the contextual foundation for the challenges addressed in this thesis.

1.4.1 Addressing the Research Questions

The following outlines how each research question is addressed through the methods, experiments, and articles included in this thesis.

RQ0: How can efficient on-device, platform-agnostic transport mode detection be achieved on mobile devices?

RQ0 serves as the overarching research question guiding this thesis and is addressed through the combined outcomes of RQ1, RQ2, and RQ3. Efficiency in on-device processing is achieved by optimising model complexity and reducing inference latency, as explored in RQ3. Platform-agnostic capabilities are realised by developing and evaluating models across multiple mobile operating systems and hardware configurations, as demonstrated in the articles underpinning RQ1, RQ2, and RQ3. Through answering this question, this thesis presents a unified approach to transport mode detection, synthesised through the underlying articles. It addresses key limitations of prior centralised and platform-specific approaches, thereby enabling practical and efficient implementation directly on mobile devices.

While Articles II–V directly contribute to answering the core research questions of this thesis, **Article I** serves a complementary and foundational role. It provides a comprehensive survey of automated fare collection systems in public transportation, identifying critical gaps and motivating the need for user-independent, smartphone-based transport mode detection.

Although Article I does not directly address the research questions, its insights guided the overall research direction and informed the problem framing for subsequent investigations.

RQ1: How can machine learning models achieve high accuracy in transport mode detection across diverse transport modes, ensuring generalisability in real-life applications?

This question is addressed through the development and evaluation of machine learning models trained on the NOR-TMD dataset, which includes real-life data collected from both Android and iOS devices, across multiple participants, devices, and transport modes. The emphasis is on achieving generalisable and robust classification performance across diverse modes of transport.

Article II establishes the feasibility of transport mode detection using real-life sensor data collected from regular travellers in Norway. It applies a multilayer perceptron (MLP) model to classify a subset of transport modes based on Android data. Rather than focusing on optimising classification accuracy, this article provides foundational insights into data preprocessing and model performance that inform later work.

Article III directly targets the goal of maximising classification accuracy. Building on the preprocessing insights from Article II, it applies extreme gradient boosting (XGBoost) to a broader and more diverse dataset. This article expands the scope to include a wider range of transportation modes and evaluates models trained separately for Android and iOS platforms. The results demonstrate that high-accuracy transport mode detection is achievable using a large and diverse dataset, in combination with carefully selected preprocessing techniques.

Collectively, these studies directly addresses RQ1 and demonstrate the feasibility of building machine learning models that maintain high accuracy and generalisability across platforms, users, and transport conditions.

RQ2: How can a standardised framework for feature evaluation and reduction systematically identify relevant features, ensuring consistency and enabling reliable feature reduction across machine learning models?

This question is addressed through multiple rounds of feature evaluation and reduction experiments, utilising a variety of machine learning models and different subsets of the

NOR-TMD dataset.

Article II investigates several feature evaluation techniques and demonstrates that they behave quite differently when applied to the same input, highlighting a lack of consistency in current approaches. The results show that an effective trade-off between accuracy and feature reduction is achieved by selecting the intersection of the top-performing features across multiple evaluation methods.

Article III adopts a fundamentally different strategy by employing a feature ablation approach, where individual features are systematically removed to assess their impact on model performance. This method reveals that features can compensate for one another, which affects the apparent importance of individual sensors. However, it proves difficult to reduce feature dimensionality in a systematic and reliable manner using this approach. As a result, it was not pursued in subsequent experiments.

Article IV builds upon the findings of Article II and introduces the EFR-TMD framework. This standardised framework for feature evaluation and reduction integrates multiple feature importance techniques to systematically rank features in a generic and model-agnostic way. The results in Article IV demonstrate consistent performance across several algorithmic approaches when feature reduction is guided by EFR-TMD.

These findings are further validated in **Article V**, where EFR-TMD is employed to significantly reduce the number of features. The final model is then deployed and evaluated on real devices operating onboard public transport vehicles, achieving satisfactory results in a real-world context.

These investigations lead to the development and evaluation of the EFR-TMD framework, which offers a consistent and reliable method for feature evaluation and reduction across different models. This outcome addresses the core aims of RQ2.

RQ3: How can transport mode detection models be optimised for real-time, low-latency inference while maintaining computational efficiency on mobile devices?

This question is addressed through both the development of the EFR-TMD framework and real-world evaluation of optimised models deployed on mobile devices.

Article IV demonstrates a direct correlation between systematic feature reduction using EFR-TMD and decreased inference times across multiple machine learning algorithms. By reducing input features, the models achieve significantly lower latency, meeting real-time processing requirements essential for on-device deployment.

Article V validates the findings from Article IV by deploying and implementing a platform-agnostic reduced-dimension model locally on Android and iOS smartphones. This model is then evaluated in real-life contexts in public transport scenarios. Local deployment eliminates network communication delays, further minimising latency and ensuring faster inference. Additionally, energy consumption measurements on Android devices reveal only marginal power usage increases during model operation, confirming that computational efficiency is maintained in practical usage.

Together, these results confirm that transport mode detection models can be effectively optimised for real-time, low-latency inference with minimal computational overhead, enabling efficient, platform-agnostic on-device operation.

Collectively, these research questions are addressed through a series of targeted experiments and frameworks that together form a unified approach to transport mode detection on mobile devices. The work integrates robust machine learning methods, systematic feature selection, and on-device optimisation to enable accurate, efficient, and platform-independent inference. The key contributions of this thesis are outlined below.

1.4.2 Contributions

The research questions outlined above have been systematically addressed through a combination of empirical studies, methodological frameworks, and real-world evaluations. This thesis integrates cross-platform machine learning, consistent feature evaluation, and on-device optimisation to overcome key limitations in existing transport mode detection approaches. The outcomes of this work are reflected in the following key contributions:

NOR-TMD: A Comprehensive Dataset for Transport Mode Detection

Development of a large-scale dataset collected from over 100 participants in two Norwegian cities. NOR-TMD includes sensor data from both Android and iOS devices and covers a wide range of transport modes, addressing gaps in existing datasets and supporting generalisable, cross-platform research.

Accurate and Generalisable Transport Mode Classification

Demonstration that XGBoost-based models trained on the NOR-TMD dataset can achieve high classification performance across diverse transport modes and device platforms, supporting real-world applicability.

EFR-TMD: A Framework for Feature Evaluation and Reduction

Introduction of an ensemble-based framework that combines multiple feature importance techniques to identify the most relevant input features. EFR-TMD is model-agnostic and demonstrates consistent performance across different machine learning algorithms and platforms.

Lightweight, Platform-Agnostic On-Device Framework

Implementation and evaluation of a transport mode detection framework, capable of classifying transportation modes on both Android and iOS devices. The framework achieves low-latency inference with minimal energy overhead, supporting practical deployment in real-world environments.

Together, these contributions form a comprehensive approach to efficient, accurate, and user-independent transport mode detection on mobile devices. By addressing challenges related to cross-platform compatibility, real-time inference, and systematic feature selection, the findings bridge existing gaps in the literature and support practical deployment in real-world public transportation scenarios. To contextualise the impact of this research, the following section situates this research within the broader landscape of Smart Mobility and Intelligent Transportation Systems (ITS).

1.5 Positioning of Research

This thesis addresses critical challenges identified in transport mode detection (TMD) research and contributes to advancing Intelligent Transportation Systems (ITS) and Smart Mobility. Smart Mobility is a broad field encompassing a variety of aspects such as sustainability, user experience, and technological innovations [22], [23]. Within its technological domain, Intelligent Transportation Systems (ITS) play a central role. As previously mentioned, this research falls within the ITS component of Advanced Public Transportation Systems (APTS).

Figure 1.2 illustrates the specific positioning of this research within the APTS component of ITS. Automated fare collection (AFC) systems are a key component of APTS and can be enhanced through the integration of transport mode detection. Within the field of transport mode detection, this thesis specifically examines challenges related to predictive accuracy, platform compatibility, feature importance, and computational efficiency, as highlighted in the blue area of Figure 1.2.

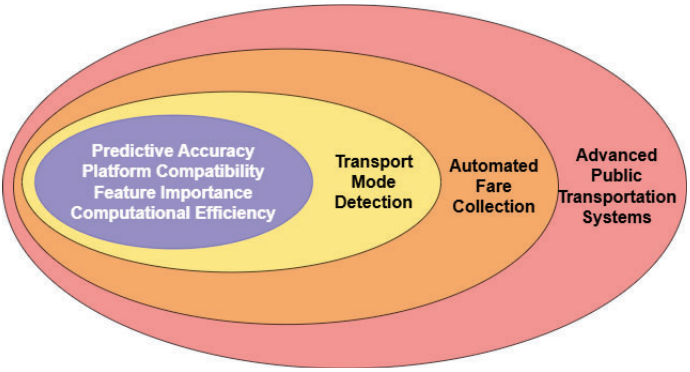


Figure 1.2: Positioning of research.

Early in the research, the focus was on understanding and improving AFC systems, culminating in Article I that thoroughly reviewed the state of the art in this domain. The findings of this review highlighted significant challenges in AFC, particularly the need for accurate and seamless in-vehicle presence detection to enable fully automated ticketing solutions. This realisation led to a shift in research focus towards transport mode detection, which can be employed to achieve in-vehicle presence detection and enhance AFC systems. However, the specific challenges associated with transport mode detection were identified through subsequent literature reviews conducted after this initial shift in focus. Therefore Article I represents a significant contribution to the field while also serving as a stepping stone for the later work presented in this thesis. This shift in focus towards transport mode detection laid the groundwork for addressing the critical challenges in achieving seamless automated fare collection.

This evolution of focus aligns with the principles of Design Science Research (DSR), a paradigm that emphasises iterative problem-solving and the development of practical artefacts to address real-life challenges. Within the DSR framework, flexibility is essential, allowing researchers to adapt their focus as new insights emerge and a deeper understanding of the problem domain is achieved. By following this approach, the research transitioned from analysing AFC systems to developing solutions for transport mode detection, a critical enabler for automated fare collection. The identification of key challenges in transport mode detection through consecutive literature reviews further refined the research direction, ensuring that the solutions developed effectively addressed the most pressing issues in the field of transport mode detection.

Building on insights from these reviews, this research addresses key challenges through the development of innovative solutions. By focusing on transport mode detection, it directly contributes to Smart Mobility and Intelligent Transport Systems (ITS), particularly within advanced public transportation systems (APTS). Leveraging machine learning and mobile computing, this research aims to achieve platform-agnostic, on-device transport mode detection. A broad range of machine learning approaches is explored to achieve high classification accuracy across diverse transport modalities using only inertial and ambient sensor data from mobile phones. The importance of various feature sets is examined, facilitating dimensionality reduction and enhancing model efficiency for mobile deployment. Additionally, platform compatibility is investigated, leading to the development of a framework capable of locally classifying transport modes across different platforms. By improving computational efficiency and cross-platform compatibility, this research reduces both the training and execution requirements of machine learning models, aligning with the sustainability goals of Smart Mobility. Furthermore, accurate transport mode detection supports Be-in/Be-out systems that operate without explicit user interaction, enhancing accessibility and inclusivity. This aligns with the efficiency objectives of ITS by facilitating faster boarding and alighting of public transportation vehicles.

1.6 Structure of the Thesis

This thesis is based on five articles, one of which are currently under review. Article I, which reviews automated fare collection systems is included as a key result of the initial phase of the research. It provides a detailed overview of current AFC solutions and their enabling technologies, laying the groundwork for the subsequent focus on transport mode detection as an enabler of advanced AFC systems.

The next chapter presents an overview of related work on transport mode detection, emphasising various approaches concerning data foundations, algorithms, and data processing strategies used to tackle challenges in this field. Chapter 3 outlines the research methodology used to address the research questions posed in Chapter 1, with a focus on the theoretical framework and quantitative methods for data collection, processing, and analysis. Chapter 4 details the findings from a series of experiments and research activities conducted during the study. This includes an in-depth presentation of the NOR-TMD dataset, the development of a generalisable feature importance framework for TMD, EFR-TMD, and the creation of a platform-agnostic framework for local TMD. Furthermore, the chapter examines the performance outcomes of various technical approaches investigated during the research, alongside the results of real-life evaluations. In Chapter 5, the results are discussed in relation to prior research, demonstrating their contributions to real-time mobility analytics, intelligent transportation solutions, and sustainable urban mobility. Finally, Chapter 6 concludes this thesis by answering the research questions, summarising key contributions and limitations, highlighting their theoretical and practical implications, and proposing directions for future work.

Chapter 2

Related Work

This research seeks to advance the fields of Intelligent Transportation Systems (ITS) and Smart Mobility by addressing the pressing challenges posed by rapid urbanisation and the increasing demand for efficient, inclusive, and sustainable public transportation solutions. The motivation stems from the need to enhance transportation systems through innovative, data-driven approaches that reduce congestion, minimise environmental impact, and improve user experiences. A central focus of this thesis lies in leveraging machine learning and smartphone sensor data to enable accurate transport mode detection (TMD). By exploring how machine learning and sensor technologies can support intelligent public transportation systems, this research aligns with broader goals of Intelligent Transportation Systems (ITS) and Smart Mobility, including the development of user-centred, privacy-preserving solutions that operate locally on mobile devices. Such advancements aim to empower individuals while contributing to sustainable and integrated mobility systems.

To position this research within the wider academic context, this chapter reviews the state of the art in machine learning-based TMD, examining algorithms, preprocessing strategies, evaluation techniques, and data resources. The review highlights existing approaches and identifies key gaps and challenges in current research, particularly with respect to accuracy, standardisation, and computational efficiency. By examining the strengths and limitations of prior work, this chapter sets the stage for understanding how this research builds upon and extends existing efforts to advance the field. Before diving into the technical details of prior studies, this chapter begins by providing an overview of the diverse application areas where transport mode detection play a pivotal role in enhancing mobility and addressing urban transportation challenges.

2.1 Overview of Applications of Transport Mode Detection

At a high level, the field of transport mode detection (TMD) can be categorised based on the diverse use cases it addresses, which are shaped by the distinct goals and practical requirements of various research domains. The primary fields where TMD finds application include location-based services, transportation science, and human geography [13].

In location-based services, the focus is on real-time identification of transportation modes, which is particularly useful in automated fare collection (AFC) systems. Notably, TMD is integral to the Be-In/Be-Out (BIBO) paradigm, where ticketing is fully automated without requiring explicit user action. Researchers have developed various solutions incorporating TMD as a component into AFC systems to detect whether a traveller is currently within a specific mode or vehicle [14], [33], [37]. Traditionally, this presence detection has relied on external hardware, such as reference devices, correlating data between the traveller's device and that of the reference device in order to place a traveller within the given vehicle [55], [81]. However, the integration of transport mode detection into AFC systems reduces the reliance on external infrastructure, thereby lowering installation and maintenance costs [14]. Furthermore, real-time contextual information derived from transport mode detection can be used for targeted advertising [77], [82], [83], such as offering discounts on gasoline to drivers [84]. The contextual information derived from transport mode detection also supports traffic congestion estimation, alternative route suggestions [83], and personalised assistance in adjusting schedules or meeting agendas based on estimated travel times for specific transportation modes [85].

In transportation science, the emphasis is on analysing and measuring daily travel patterns of individuals or groups over historical time frames [13]. The most common application of transport mode detection in this field is the automation of travel surveys [82], [86]–[89]. Detailed travel information is critical for understanding individual travel behaviour and supporting transportation planning decisions [90]. Traditional methods for collecting this kind of data, such as manual questionnaires and telephone surveys, often result in under-reporting of short trips and inaccurate or incomplete data [86], [91]. Transport mode detection offers significant potential to mitigate these limitations by minimising biased responses, reducing instances of non-responses, and improving the accuracy of time reporting [89] by automatically estimating travel information, without the need for manual questionnaires or telephone surveys. Insights into transportation modes provide valuable information for public transport operators aiming to encourage shifts from private cars to public transit. Similarly, transport providers can analyse urban mobility trends to optimise services [76]. It also supports the estimation of origins and destinations (O-D) to quantify transport demand between city regions, enabling evaluation and planning of traffic, identification of optimal new transport routes, and analysis of trip purposes and weekly travel patterns [46], [92], [93]. Additionally, transport mode detection is increasingly used to estimate environmental impacts, including carbon footprints and emissions associated with different transportation modes, reflecting its growing importance in addressing environmental concerns [82], [94].

In the field of human geography, transport mode detection (TMD) is instrumental in enriching trajectories by incorporating domain-specific semantic information [13]. TMD enables the segmentation of trajectories into meaningful parts, distinguishing between stationary and mobile segments. This segmentation facilitates further semantic enrichment, such as identifying activities associated with points of interest, changes in travel direction, and temporal movement patterns [13]. By analysing transitions between stationary and mobile states, human geographers can study spatial behaviour and decision-making processes related to movement. Combined with auxiliary datasets, transport mode detection can support human geography research in diverse contexts, including migration studies [13] and urban planning [95]. Beyond location-based services, transportation science, and human geography, transport mode detection also has applications in health monitoring [45], [82], [96] and by determining transportation modes, it can estimate daily activity patterns and caloric expenditure, providing insights into individuals' physical activity levels [82], [96].

Having briefly explored the diverse application areas of transport mode detection, attention is now turned to the different approaches that form the foundation of these applications, examining how different techniques are employed to achieve effective transport mode detection.

2.2 Approaches for Transport Mode Detection

Achieving accurate transport mode detection (TMD) primarily relies on three methodological approaches: network-based, location-based, and sensor-based. Each of these approaches leverages distinct data sources and analytical techniques, offering unique advantages and challenges in different application contexts. In addition, hybrid methods have emerged, integrating data sources and analytical techniques from all three approaches to enhance detection accuracy and contextual adaptability. Figure 2.1 presents an overview of the three primary approaches to transport mode detection and demonstrates how hybrid methods combine their respective data sources. These approaches have evolved to address the diverse requirements of TMD applications, ranging from real-time responsiveness to historical analysis and semantic trajectory enrichment. By utilising various data sources, such as network infrastructure, geographical positioning, sensor readings, or combinations thereof, network-based, location-based, sensor-based, and hybrid methods cater to the specific goals of different research domains. In the following subsections, each approach is explored in detail, highlighting their key characteristics, advantages, and limitations.

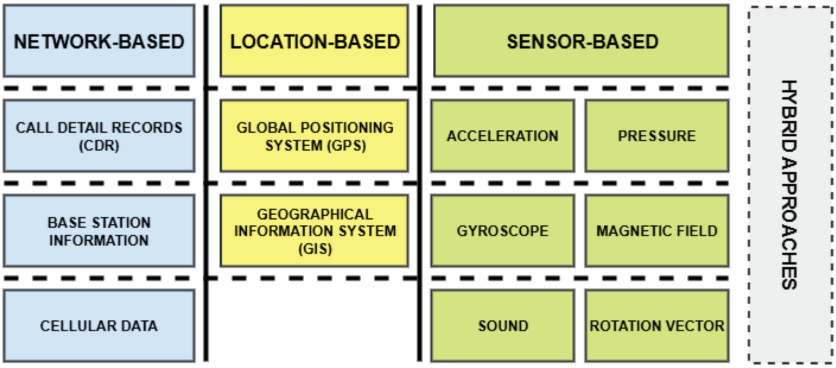


Figure 2.1: Overview of Approaches for Transport Mode Detection

2.2.1 Network-Based

Network-based approaches utilise coarse-grained network data such as call detail records (CDR), cellular data or base station information [71], [76], [96]–[104]. CDR typically includes the numbers of the caller and callee, start time, duration, the exchange identifier and similar data points. Additionally, information about the cell tower the device is connected to and an approximate location is also provided [97]. This approximate location is what is used to estimate trips. In order to estimate whether a traveller is using private or public transportation, the travel time between each origin and destination can be correlated with the average travel time provided by solutions such as Google Maps. Trips with a travel time close to the average travel time using public transportation are then likely to be trips conducted onboard public transportation. This works equivalent when estimating private transportation [97]. CDR data is cheap, as telecom equipment already generates this data when users are making phone calls and send/receive messages. CDR data can also be used to separate road and rail travel by labelling data that comes from base stations and equipment known to be situated in the underground or train stations [71]. A problem with this approach is that it requires people to make frequent calls or send/receive messages [97]. While people might be more prone to send messages and call during public transportation, they might be less likely to do so when driving private vehicles, which makes it difficult for this approach to establish granular modes [71]. That being said, given enough data points it is possible to detect more granular modes such as train, subway and walking by extracting features representing motion based on the location data within the records.

By mapping the network data onto nearby roads, using data from geographic information systems (GIS), some researchers have been able to separate the modes car and bus [103], however modes such as bike and tram are still challenging in urban areas [99]. Although the use of coarse-grained network data is mainly useful when investigating travel patterns in historical contexts, some researchers have proposed a solution for real-time network-based transport mode detection by using the received signal strength (RSS) and the cell identifier (CID) [104], however only to distinguish between stationary, walking and motorised travel. In order to increase the number of possible modes to classify, Cardoso, Madureira, and Pereira [96] employed data from wireless access points (WAP) to classify modes of transportation. This approach establish a rough location estimate based on the WAP data, which in turn is used to train a classifier, outperforming existing off-the-shelf activity recognition libraries, such as Google Activity Recognition API and Intel Activity Recognition [96]. While network-based approaches demonstrate potential, they typically support fewer transport modes and exhibit lower accuracy compared to location- and sensor-based approaches. Additionally, network-based methods often depend on travellers actively using their devices or being in close proximity to wireless access points. To detect granular modes of transportation in real-time, with high accuracy to enable solutions such as AFC and real-time information systems, fine-grained data such as GPS coordinates or smartphone sensor data is needed.

2.2.2 Location-based

This brings us to location-based solutions, which leverages Global Positioning System (GPS) data, either by itself or in combination with data from Geographical Information Systems (GIS) to achieve accurate detection of transportation modes [77], [83], [87], [105]–[118]. As with all research and development of machine learning technology, large amounts of GPS data is required to achieve accurate location-based solutions. In the early days of smartphones, assisted GPS (AGPS) service providers imposed limitations on GPS sampling frequencies of mobile devices, and because of this many early works investigating the use of GPS for transport mode detection employed separate GPS loggers to record coordinates across various transportation modes using different sampling frequencies [105]. When employing GPS data to estimate the mode of transportation, researchers mainly rely on the speed and acceleration, which are derived from the location data [119]. In addition, some researchers have developed new features such as bearing rate, change in heading, and distance travelled calculated from the GPS coordinates [53], [106], [107], [109]. From these features, speed and acceleration has been deemed the most useful features to infer the mode of transportation [105], [106] and are present in most works concerning location-based TMD solutions. From speed and acceleration, new features can also be extracted by applying statistical functions, such as calculating the average or the percentiles of speed, which further improves classification accuracy [107].

As smartphones became more widely adopted, researchers started to use the built-in GPS module of mobile devices, and instead of having participants carrying dedicated GPS loggers, they could instead develop and install custom mobile applications that continuously collected GPS data in order to amass larger datasets for training and validating their research [107], [109]. This enabled researchers to maintain a better control over the process of labelling the data as well as providing flexibility in terms of sampling rates, where higher sampling frequencies can lead to more accurate solutions [105]. Past research efforts aimed at achieving location-based TMD solutions using only GPS data have managed to increase the granularity of modes, compared to network-based approaches, and have managed to classify when travellers are walking, riding bikes or e-bikes, taking the bus, driving a car or using various rail transportation such as metro, tram or train [107]–[109], [120]. However, GPS can be unreliable when travelling underground or in dense urban canyons since GPS require a clear view of the sky in order to establish accurate locations [60], [77], [121].

In order to address these limitations authors have incorporated GIS data, which includes data on road networks, subway networks, train networks, bus stops, and stations [105], [115], [122]. Studies have shown that incorporating GIS data can enhance the classification accuracy significantly, compared to using GPS data alone [77], [115], [122]. By leveraging GIS data it is possible to achieve route-specific classifications, where the possible vehicles that frequent a specific route is taken into account [105]. Route-specific classification schemes have exhibited better classification accuracies as opposed to general mode classification [105], however systems incorporating GIS, are more complex and less robust to changes in the urban environments, such as changes in transportation infrastructure (e.g., new routes and roads).

While incorporating GIS data can enhance the accuracy of location-based approaches [77], [115], high classification performance can also be achieved using only GPS data [107]. Despite their success, location-based methods face several limitations. First, GPS requires an unobstructed view of the sky, making it unreliable for underground travel or in dense urban environments such as urban canyons [60], [77], [121]. Second, GPS is highly energy-intensive, posing challenges for resource-constrained devices like smartphones [60], [77], [123]. Third, the use of GPS trajectory data raises significant privacy concerns. GPS data can be cross-referenced with external datasets to identify individuals, even with a small number of unique location points. This raw data can reveal sensitive information, such as home and work locations, personal movement patterns, and lifestyles. Moreover, frequent visits to specific locations, such as religious or political venues, can expose personal beliefs and affiliations [66]. Given these limitations, many researchers have shifted their focus to sensor-based approaches, which leverage the rich data available from smartphone sensors to address challenges related to GPS reliability, energy consumption, and privacy concerns.

2.2.3 Sensor-based

In order to combat the limitations of GPS-based methods, sensor-based approaches harness smartphone sensor data to improve accuracy and overcome challenges associated with energy consumption and privacy. Sensor-based approaches differentiates from location-based approaches in that they employ inertial and ambient sensor data generated by the hardware of mobile devices. Mobile sensor data is converted into transportation information by analysing unique signatures within each travel mode such as rolling, waving, vibration, and acceleration patterns [124]. This data is then used to train machine learning algorithms to classify the mode of transportation, similar to that of location-based solutions. Within the realm of sensor-based solutions, acceleration is the most widely applied feature for sensor-based transport mode detection solutions, but rather than calculating an approximate acceleration or speed, from potentially unreliable GPS coordinates, sensor-based approaches instead sample the built-in accelerometer in mobile devices [60], [79], [125], [126]. By employing only data from the accelerometer, researchers have been able to achieve promising results, classifying modes such as walking, riding a bike, riding the bus, metro, or train, and driving a car or motorcycle with decent accuracy [60], [79], [121], [125]–[127]. However, while accelerometer data is less intrusive than location data, it can still present privacy concerns, as it may reveal insights into a user's movements, identity, demographics, and activities, especially when combined with other data sources [67].

As mobile devices are becoming more advanced, new smartphones now have a plethora of ambient and inertial sensors, such as magnetometers and gyroscopes, which can be leveraged to improve the classification accuracy of TMD solutions. The gyroscope presents the angular velocity of rotation along three axes [128] and can provide other dimensions of patterns related to different modes of transportation. For instance, the gyroscope can be leveraged to find unique signatures in how the vehicles vibrates [129], leading to the possibility of discerning when a vehicle has stopped due to congestion or red lights, as opposed to having parked or the traveller has disembarked. Similarly, the magnetometer measures the magnetic field around the device across three axes [128] and is altered by the vehicle's structure, creating a distinct signal that can be used to identify a person's mode of transportation [85]. Different combinations of accelerometers, magnetometers and gyroscopes have been widely deployed in different algorithmic approaches with solutions exhibiting accuracies over 90% [85], [130]–[134]. Although, somewhat less applied in research, the barometer is also an important sensor in the context of TMD solutions. As opposed to the accelerometer, magnetometer, and gyroscope, the barometer is independent of phone position and orientation, and was originally included in mobile devices to reduce the delay of the GPS fix by providing altitude [135]. However, the barometer is sensitive enough to detect height differences when moving along a road, even though the differences are too small to observe by the naked eye, making it suitable for generating data consistent with different modes of transportation, while simultaneously using very little energy [135]. The barometer have been used in combinations with other sensors, such as accelerometer, gyroscope, and magnetic field, in order to further enhance solutions for sensor-based TMD solutions, exhibiting accuracies well beyond 90%, highlighting the importance of this sensor [136]–[138].

Building on this, researchers have applied every standardised sensor that exists on modern mobile devices, including fused sensors such as the different rotation vectors, linear acceleration, and gravity [74], [84], [136], [139], [140]. Going beyond that of ambient and inertial sensors, some researchers have investigated the applicability of sampling sound from mobile devices in order to classify the mode of transportation [127], [140], [141]. Sound is a key modality accessible on all smartphones via the microphone, and can be more effective for classifying vehicular modes compared to motion sensors [141]. This is because each vehicle typically produces a distinct sound, making it easier to distinguish vehicles that exhibits similar sound patterns [141]. Although sound can serve as a valuable and effective data source for developing solutions for transportation mode classification, it introduces significant privacy concerns that must be carefully addressed [127], [140], [141]. In order to address privacy concerns associated with using sound, researchers have proposed techniques such as selectively sampling audio with intermittent muting, ensuring the captured sound remains incomprehensible while still achieving comparable performance for transportation mode detection [127]. More broadly, sensor-based approaches have shown great promise in transportation mode detection. However, to further improve performance, some researchers have explored combining methods. Hybrid approaches leverage the strengths of multiple methodologies to enhance the accuracy, robustness, and reliability of transportation mode classification.

2.2.4 Hybrid Approaches

Many approaches combining both location-based and sensor-based approaches, employ the combination of GPS and acceleration data [56], [89], [142], [143]. Accelerometer and GPS data complement each other because they capture different aspects of movement. When accelerometer outputs appear similar across modes of transport, the GPS speed can help distinguish them, and vice versa; when speed data is similar, differences in acceleration patterns can provide clarity [142]. Reddy, Burke, Estrin, *et al.* [142] states that removing either of the two features caused a significant accuracy drop in their experiment, highlighting the necessity for combining the two approaches. Building on this approach, researchers have proposed using two different classifiers, to separate when travellers are stationary from when they are moving, within modes such as driving, walking, or riding a bike [129]. Shah, Wan, Lu, *et al.* [144] proposed a system that utilise multiple classifiers to enhance this methodology. At the initial stage, a classifier based on acceleration data is employed to differentiate less granular transportation modes, such as stationary and motorised states. Upon detecting vehicular motion, a higher-level classifier, incorporating GPS and GIS data, is applied to classify more specific modes of transportation with greater precision. An important aspect is being able to separate stationary and motion within different modes because even though travellers might be situated within a vehicle, if the vehicle is stationary over a longer period it can be difficult to classify the correct mode, due to the dependency on speed and acceleration. However, separating stationary and moving within different modes is difficult and the two classes are easily confused when the solution is based on only location data and accelerometer data [129].

Further advancing hybrid approaches, researchers have proposed to combine GPS and acceleration with data from the magnetometer [64], [94], [145], [146], as the magnetic field is particularly useful when classifying modes such as tram and train which cause significant magnetic disturbances [94]. From there, a plethora of sensor combinations together with location data, such as GPS and GIS, have been assessed, including gyroscopic data, rotation vectors and sound [14], [66], [141], [147]. Many hybrid approaches successfully leverage the complementary strengths of location-based and sensor-based methods to achieve higher accuracy and granularity in transportation mode detection. By integrating diverse data sources and employing multiple classifiers, hybrid methods address limitations inherent to individual approaches, such as distinguishing stationary states or classifying similar modes like tram and train. That being said, by employing location data, hybrid solutions still face the same challenges of energy consumption and privacy as location-based approaches.

2.3 Machine Learning in Transport Mode Detection

Machine learning has become a cornerstone of transport mode detection (TMD), offering better generalisation and improved discrimination when classifying transportation modes, compared to traditional approaches [54]. Both supervised and unsupervised machine learning methods have been explored to model patterns in sensor data, each suited to specific contexts and requirements. However, the success of the different methods relies heavily on effective preprocessing to prepare the raw data and the evaluation frameworks used to assess model performance. This section reviews the key machine learning approaches applied to TMD, the processing techniques used to optimise data for analysis, and the methods for feature selection, extraction, and importance.

2.3.1 Algorithmic Approaches

Most machine learning approaches for transport mode detection employs a supervised approach. Supervised learning techniques rely on labelled data to learn patterns and relationships that enable the classification of past or current events. During the training process, supervised algorithms develop an inferred function that maps input data to the corresponding output values. With sufficient training, the model becomes capable of accurately predicting outcomes for new, unseen data. Supervised learning algorithms iteratively compare the predicted outputs with the actual labels, identifying errors and adjusting the model to minimise discrepancies. This process ensures continuous refinement and improved accuracy in forecasting tasks [148].

Supervised learning is widely applied within the field of TMD and a large variety of supervised algorithms have been utilised towards achieving accurate TMD solutions. Traditional algorithms such as Bayesian inference, k-nearest neighbours (KNN), support vector machines (SVM), decision trees, and random forests have been widely utilised [56], [62], [64], [71], [82], [96], [119], [143], [149]. Decision trees are particularly notable for their interpretability, as they mimic the human decision-making process and can be easily visualised. However, large decision trees often suffer from overfitting [94], a common challenge in supervised learning where the model fails to generalise to unseen data [150]. To address this limitation, random forests extend the decision tree concept by building an ensemble of multiple trees, each trained on a randomly selected subset of features. The final prediction is determined by a majority vote across all trees, improving robustness and reducing overfitting [120]. Advanced and emerging approaches have further expanded foundational methods, including techniques like gradient boosting [57], [80] and stacked learning [74]. Boosting is a technique used to reduce bias and variance [148] and forms the basis for algorithms such as adaptive boosting (AdaBoost) [89] and extreme gradient boosting (XGBoost) [151] which also have been utilised in the context of TMD with promising results.

Traditional algorithms often struggle with high-dimensional data and tend to be sensitive to irrelevant features. In contrast, neural networks present a robust alternative, overcoming many limitations of conventional methods, which typically rely on strict assumptions like normality, linearity, and variable independence [152]. Multilayer Perceptron (MLP), which is a feed-forward neural network (DFNN), have been employed in this context [37]. However, DFNNs cannot retain information over time, but instead analyse each sample independently. This limitation has led to the exploration of neural network architectures capable of processing sequential data, such as gated recurrent units (GRU) and long short-term memory (LSTM) [75], [138], [153]. Additionally, some researchers have utilised convolutional neural networks (CNNs) in the context of TMD [117], [154]. Although CNNs are predominantly used in image and video recognition due to their ability to extract hierarchical spatial features, their architectural design limiting connections to local receptive fields also reduces the risk of overfitting [155], making CNNs a viable option for various classification tasks. In the context of both human activity recognition (HAR) and TMD, combining LSTM and CNN architectures has also been explored to leverage both temporal and spatial feature extraction capabilities, showing potential for enhanced performance in sequential classification tasks [137], [153], [156].

While the majority of studies employ supervised learning approaches using collected or pre-existing datasets, some researchers have proposed semi-supervised or unsupervised methods for classifying modes of transportation. In contrast to supervised methods, unsupervised methods do not require labelled data. Instead, they aim to extract inferences from the input data, modelling the hidden or underlying structures and distributions within it [157]. Researchers have employed different forms of clustering in order to estimate the mode of transportation [99], [158]. Another commonly used unsupervised approach is hidden Markov models (HMM) [86], [149], [159].

Supervised and unsupervised approaches have also been combined, forming semi-supervised approaches [111], [116]. These approaches consist of multiple parts where the unsupervised component captures general patterns and structures in the data from both labelled and unlabelled datasets, enriching the feature representations, while the supervised component leverages the labelled data to perform precise classification tasks. Estimating transportation modes based on GPS data, Dabiri, Lu, Heaslip, *et al.* [111] utilised a combination of CNN and a convolutional-deconvolutional autoencoder, which is an unsupervised learning technique that aims to learn an efficient latent representation by reconstructing the input at the output layer. While this approach reduce the need for labelled data, the approach struggles with discriminating between similar modes, especially when the labelled data for the different modes is limited. On the other hand, Zhang, Zhu, Markos, *et al.* [116] combined pseudolabelling with an ensemble of neural networks, within a federated learning framework in order to reduce the need for labelled data, while simultaneously preserving user privacy.

While numerous studies examine the strengths and weaknesses of various machine learning algorithms [148], [152], [155], [160]–[163], direct comparisons remain challenging due to the lack of standardised benchmark datasets [80]. The performance of any given algorithm is highly dependent on the data it is trained on, making it difficult to draw definitive conclusions about the superiority of one approach over another. Given this variability, the effectiveness of a machine learning model is not solely determined by the choice of algorithm but also by the quality and structure of the input data. Proper preprocessing, including segmentation, transformation, and feature extraction, is essential to maximise model reliability and efficiency.

2.3.2 Data Preparation and Segmentation

In practice, raw data must be carefully prepared to unlock the full potential of machine learning algorithms. While a wide range of algorithms is available, an even greater variability exists in the techniques used for data preparation and feature extraction for training machine learning models. One well-established approach, the Activity Recognition Chain (ARC) framework [59], outlines the steps required to infer user activity from inertial sensors in human activity recognition. This framework highlights the importance of segmentation and recommends a sliding window approach for segmenting time-series data. Although ARC is primarily focused on activity recognition, sliding windows are also widely used in transport mode detection preprocessing [60], [96], [164], [165]. Window functions consist of a size and overlap, and by continuously stepping through the data, sliding windows provide a sequential structure that reveals transient trends and enhances pattern detection accuracy. The size of sliding windows typically ranges from 1 to 60 seconds [14], [73], [85], [126], though longer windows can also be effective [73], [98], [135]. However, there is a trade-off between finer granularity with increased window density and the computational demands of shorter windows [73]. Some researchers have also proposed dynamic sampling windows that can cover an entire travel period and automatically adjust based on the individual's activities [124].

When processing data from three-dimensional inertial sensors, many studies compute the magnitude of the three axes to remove directional dependencies before aggregating segments [96], [143], [164]. However, other studies choose to retain directional values [14], as they provide more nuanced information. The window size is also influenced by the initial sampling frequency since higher sampling frequencies allow for more data points within each window. Sampling frequencies used in previous studies vary widely, ranging from 1/90 Hz [99] to 100 Hz [85], [89]. Research involving network or location data in historical contexts typically employs much lower sampling frequencies (less than 1 Hz) [99] compared to studies focusing on real-time sensor data [82], [85], [89], [99], [126], [133], [146], [147]. The sampling frequency significantly affects both the accuracy and efficiency of the methodology. While lower frequencies reduce accuracy, they also decrease computation time, demonstrating a trade-off that must be carefully considered [166]. To ensure a uniform sampling frequency or modify the initial frequency, interpolation techniques have been employed [14], [125], [135].

When extracting features, both time-domain and frequency-domain features are widely utilised in transport mode detection research. Time-domain features capture the signal's characteristics as it varies over time, while frequency-domain features can offer additional insights but are considered more computationally demanding due to the required transformation step [62]. Transforming data into the frequency domain typically involves applying a variant of the Fourier transform, most commonly Fast Fourier Transform (FFT) [79], [125], [127], Discrete Fourier Transform (DFT) [143], or Short-Time Fourier Transform (STFT) [73], [141], [147]. Although frequency-domain transformations are frequently employed in transport mode detection, many researchers prefer using time-domain features [14] or combining both domains [60], [70], [102], [104], [167]. Once the raw data is converted to the desired domain and segmented into windows, the values within each segment must be aggregated into representative features to ensure that each window provides a concise input for model training. Various statistical functions have been used for data aggregation, including variance [164], mean [64], [76], [96], [119], median [96], minimum and maximum values [64], [96], [119], kurtosis [96], standard deviation [64], percentiles [168], quartiles [165], and interquartile range [57].

While the aforementioned preprocessing steps establish the foundation for most prior research efforts, many researchers face challenges with imbalanced data, where the distribution of samples across different classes is uneven. This can lead to classifier bias toward majority classes, reducing predictive accuracy for under-represented modes [169]. This issue, is commonly mitigated using the Synthetic Minority Over-sampling Technique (SMOTE) [170], which generates synthetic data for minority classes [73], [85], [118], [129]. Although less common, some studies have also explored generative adversarial networks (GANs) to address class imbalance in transport mode detection, yielding promising results [112]. The impact of preprocessing becomes evident during model evaluation, where the quality of prepared data directly affects accuracy, robustness, and reliability. However, as datasets grow in complexity, high-dimensional feature spaces can introduce redundancy and computational overhead. To address this, researchers employ dimensionality reduction techniques to refine feature sets, improving both efficiency and classification performance.

2.3.3 Dimensionality Reduction and Feature Importance Estimation

Feature selection and feature extraction are fundamental techniques for dimensionality reduction in machine learning. Feature selection involves identifying and retaining only the most relevant input dimensions for solving a given problem, while feature extraction transforms the input space onto a lower-dimensional subspace that preserves the essential information [171]. Feature selection and extraction, either used independently or in combination, enhance various aspects of machine learning models, including predictive performance, visualisation, and interpretability [172]. Features are often categorised as relevant, irrelevant, or redundant [172], with redundant and irrelevant features negatively impacting computational efficiency and model accuracy. Feature selection provides several advantages: reducing dimensionality, limiting storage requirements, improving algorithmic efficiency, eliminating redundant or noisy data, enhancing predictive accuracy, and offering insights into the data-generating process [173]–[175].

Feature selection can broadly be classified into three categories; filter, wrapper, and hybrid methods [172]. Filters apply statistical techniques to select features independently of the learning algorithm, making them computationally efficient but less reliable in classification tasks. Wrappers, on the other hand, optimise feature selection for a specific classifier by evaluating different feature subsets, leading to improved classification performance at the cost of increased computational expense, particularly for high-dimensional datasets. Hybrid methods integrate statistical tests with classifier-based evaluations, combining the strengths of filters and wrappers [172]. Among wrapper methods, sequential forward selection (SFS) is effective when the optimal subset consists of a small number of features, but struggles to remove features that become obsolete after additional selections. Sequential backward elimination (SBE), which removes features iteratively, also lacks the ability to reassess previously discarded features [172].

Handling high-dimensional data remains a challenge due to increased computational costs and memory usage [176]. Empirical studies suggest that redundant features degrade both speed and accuracy, reinforcing the need for feature selection methods that simultaneously address redundancy and irrelevance [172]. Principal Component Analysis (PCA) is among the most widely used feature extraction techniques [172]. As a non-parametric method, PCA reduces redundancy and noise while extracting the most relevant information from a dataset. However, its reliance on linear relationships among variables and the necessity for numerically scaled data limit its applicability [172]. Additionally, PCA's effectiveness varies with different data types [176], and does not always enhance classification performance [153]. A study by Subasi, Ghosh, Manzano, *et al.* [177] compared various low-cost feature reduction algorithms and found that ANOVA and PCA outperformed Gaussian and Sparse Random Projections in improving classification accuracy. Despite these advances, no universally accepted method for feature selection or importance has been established [172].

Feature importance plays a crucial role in both feature selection and model interpretability but remains highly unreliable [178]. Different machine learning models and importance techniques often yield varying importance rankings for the same dataset, complicating interpretability and trust in model outputs [175]. Moreover, there is no consensus on the most appropriate metric for evaluating feature contributions, which poses a significant challenge in safety-critical applications where explainability is paramount [179]. Model-specific techniques, such as impurity-based measures in tree-based models, often fail to generalise across different learning algorithms. In contrast, model-agnostic approaches, including SHapley Additive exPlanations (SHAP) and permutation importance (PI), attempt to provide more general estimates but still produce varying results depending on the dataset and learning model [178]. The reliability of feature importance estimates can be compromised if only a single method is used [179].

To address this, Rengasamy, Rothwell, and Figueredo [179] introduced an ensemble framework that aggregates results from multiple machine learning models and feature importance techniques, demonstrating that combining outputs improves robustness over individual methods. Their approach employs fusion metrics such as mean, median, majority voting, and rank correlation, with majority voting producing the most stable results across multiple datasets. Notably, their findings indicate that noise in datasets does not affect the ability of the ensemble method to accurately assess feature importance. Expanding on this approach, Rengasamy, Mase, Kumar, *et al.* [178] developed the FEFI (Fuzzy Ensemble Feature Importance) framework, designed to address the uncertainty and unreliability of existing post-processing feature importance techniques while mitigating ethical concerns surrounding misleading output interpretations. This framework integrates fuzzy logic and incorporates the distribution of feature importance values to establish flexible boundaries, making the results more interpretable and accessible to non-experts. However, their study relied on synthetic data, and the authors emphasise the need for further evaluations using real-life datasets to assess performance in practical applications [178].

Feature reduction is increasingly recognised as a key factor in managing the growing computational and energy demands of machine learning models [180]. Reducing the number of features leads to a significant decrease in model size [177], [181] and has been shown to reduce energy consumption and carbon emissions by 23–99% while maintaining accuracy [174].

Hestness, Narang, Ardalani, *et al.* [181] demonstrated that model size scales sublinearly with data size, reinforcing the importance of feature selection for computational efficiency. Feature reduction not only eliminates unnecessary features but can also enhance classification performance [153], [172]. However, the estimation of feature importance remains unreliable [178]. Different machine learning models, importance techniques, and data subsets, generate disparate importance coefficients, often with varying magnitudes for the same features [175], [179]. Additionally, there is no consensus on the optimal metric for feature importance calculation, highlighting the need for more reliable and accurate estimation methods [179]. This challenge extends to the selection of training data, as the quality and representativeness of datasets play a crucial role in shaping model performance.

2.4 Publicly Available Datasets

Datasets not only provide the foundational data for model development but also define the scope, generalisability, and limitations of transport mode detection (TMD) systems. A key challenge in this field is the lack of benchmark datasets, which hinders the ability to compare results across studies [80]. Although some publicly available datasets exist, they each suffer from specific limitations. In this section, an overview of publicly available datasets commonly used in TMD research is provided, focusing on their characteristics, strengths, and constraints. Many researchers develop custom applications and recruit participants to collect sensor and location data for experimentation [94], [107], [109]. However, a significant number of studies utilise publicly available datasets, such as Microsoft's Geolife dataset for location-based approaches [182]. This dataset comprises extensive location data spanning multiple transportation modes, users, and geographical locations. While smartphone sensor-based datasets are also available, they face notable limitations.

To the best of available knowledge, only four publicly accessible datasets have been systematically collected across various transportation modes: the HTC Transport Mode Dataset [123], the Sussex-Huawei Locomotion (SHL) Dataset [183], the US-TMD Dataset [72], and the Collecty Dataset [184]. The HTC Transport Mode Dataset contains 8,311 hours of data collected from 150 students and 74 employees and interns, totalling 100 gigabytes (GB) [123]. However, the data was collected through only two predefined routes, limiting its geographic diversity and generalisability. Although the dataset is extensive in terms of participant count, data size, and transportation modes, it is limited by the lack of sensor and device diversity.

Table 2.1: **Overview over publicly available sensor-based datasets:** DL = Placement of the device during data collection. N.UD = Number of unique devices. N.UP = Number of unique participants.

Dataset	Modes	Sensors	DL	N.UD	N.UP	Hours
HTC [123]	Still, Walk, Run, Bike, Motorcycle, Car, Bus, Metro, Train	Accelerometer, Magnetometer, Gyroscope	N/A	1	224	8311h
SHL [183]	Still, Walk, Run, Bike, Car, Bus, Metro, Train	Accelerometer, Magnetometer, Gyroscope, Orientation, Gravity, Linear acceleration, Ambient pressure, Google's activity recognition API, Ambient light, Battery level and temperature, Satellite reception, WiFi reception, Mobile phone cell reception, GPS, Audio	Hand, Hips, Torso, Backpack	1	3	2812h
US-TMD [72]	Still, Walk, Car, Bus, Train	Accelerometer, Magnetometer, Gyroscope, Gravity, Ambient light, Ambient pressure, Audio, Proximity	N/A	11	13	31h
Collecty [184]	Walk, Run, Bike, Car, Bus, Train, Tram, e-scooter	Accelereometer, Magnetometer, Gyroscope, Linear acceleration	N/A	Unknown	15	242h

In contrast, the SHL Dataset was collected over seven months by only three participants, resulting in 950 GB of data across 2,812 hours [183]. The SHL dataset incorporates data from 15 different smartphone sensors but was collected using a single device type, which may restrict its generalisability despite its larger scale and sensor variety compared to the HTC dataset. The US-TMD Dataset offers data from 13 participants using 11 different devices, totalling approximately 32 hours and 3 GB [72]. While smaller in scale than the HTC and SHL datasets, its inclusion of multiple device types enhances diversity. Similarly, the Collecty Dataset consists of data collected from 15 users over five months, amounting to roughly 242 hours of data [184]. An overview of the datasets is presented in Table 2.1.

While these datasets provide valuable resources for ITS and Smart Mobility research, they each have notable limitations. For instance, none of these datasets include sensor data from iOS devices, which limits their applicability across operating systems. Furthermore, they do not cover transportation modes such as seagoing vessels, which are an integral part of public transport in many countries [62]. Additional limitations include restricted diversity in sensors, platforms, devices, participants, and geographical locations. Variations in hardware, operating systems, and manufacturer-specific sensor designs lead to inconsistencies in sensor data quality and variability across different devices [48], [185]. Similarly, travel behaviours and movement patterns differ among individuals and locations and can be influenced by demographic factors [186], geographical locations [187] and personal habits [188]. As such, Gong, Zhong, and Hu [189] states the importance of diversity in training data for creating machine learning models that generalise effectively across various devices, environments, and individual behaviours. Current datasets often lack sufficient variability in device placement, participant movement patterns, and human factors such as posture, which are essential for robust transport mode detection systems. Addressing these gaps is crucial for advancing Intelligent Transportation Systems (ITS) and Smart Mobility solutions.

2.5 Implementation and Evaluation

While the development of transport mode detection (TMD) models relies on diverse datasets and advanced machine learning techniques, real-life applicability ultimately depends on how well these models perform in practical settings. Beyond achieving high accuracy on curated datasets, models must be validated, optimised, and deployed in ways that ensure reliability, efficiency, and scalability. This section explores the key aspects of TMD implementation and evaluation, beginning with model validation techniques to assess generalisation, followed by deployment and evaluation considerations related to real-life implementations.

2.5.1 Model Validation

Model validation plays a vital role in ensuring that the trained classifier performs effectively on unseen data. However, comparing the classification performance of related work is highly complex due to significant variations in the data used for training [80], as well as discrepancies in the classes that the classifiers are trained on. While some studies distinguish only between motorised and on-foot modes [104], others group different classes together, such as train and tram [124]. Some implementations consider only a limited number of transport modes, whereas others incorporate a broader spectrum, including stationary, bike, walking, car, train, tram, subway, and bus [64], [190]. A few studies even extend their scope to include seagoing vessels [191], though this remains relatively uncommon. While several studies report impressive classification accuracies exceeding 90% [14], [121], [129], [130], [134], [136], [192], many of these works rely on location data [14], [129], [130], [192]. Few studies achieve comparable accuracies using a purely sensor-based approach [74], [85], [121], [136], [140], and among these works, even fewer reach above 90% accuracy when classifying a broad range of modes [85], [121], [137]. As such, most sensor-based studies report overall classification accuracies below 90% [60], [124], [125], [131], [135], [141].

To evaluate classification performance, the most common approach is to split the dataset into training and test subsets [75], [113], [128], [131], [134], [140], [190], [193]. This approach ensures that the models are evaluated on unseen data, not present during training. Many studies also use leave-one-out validation [60] or K-fold cross-validation [113], [125], [143], [147]. Leave-one-out cross validation is most suited with working with small datasets or with limited instances per class, however when the amount of data is larger, K-fold cross validation should instead be employed [194]. That being said, employing a holdout validation by splitting the dataset is generally suggested when working with very large datasets [195].

2.5.2 Implementation and Deployment Considerations

While classification accuracy remains the predominant metric in evaluating machine learning models, inference time is increasingly recognised as a critical consideration, particularly in real-time and resource-constrained applications [196]. Several studies have explored the inference time of machine learning models to better understand the trade-offs between computational efficiency and predictive performance [14], [73], [131], [154], [192], [196]. For example, Fang, Liao, Fei, *et al.* [131] analysed the inference times of decision tree, support vector machine (SVM), and k-nearest neighbours (KNN) classifiers. Their findings demonstrated that while the SVM achieved the highest accuracy, it also incurred the highest inference time. Conversely, the decision tree, although less accurate, exhibited significantly lower inference time. The measured inference times ranged from 0.69 to 9715.80 microseconds, reflecting the diverse computational requirements of the different models. Inference time has also been studied in the context of mobile devices, where computational resources are often limited. Oplenskedal, Taherkordi, and Herrmann [14] evaluated their solution on various mobile devices, reporting inference times between 32 and 59 milliseconds (ms). Similarly, Matthes and Springer [73] investigated the impact of model configurations on inference times of convolutional neural networks (CNNs) deployed on mobile devices. Their findings highlighted a wide range of inference times, from 6.53 to 190.96 ms, depending on the use of GPU or CPU and the specific model configuration.

The relationship between model complexity and inference time has been a recurring theme in the literature. Tang and Cheng [192] observed that increasing the number of convolutional layers generally improves performance but results in longer inference times due to the increased model complexity, reporting inference times ranging from 1 to 2.3 ms. Moreau, Vassilev, and Chen [154] also explored the trade-offs in deploying larger models on embedded devices and demonstrated that while larger models can achieve higher accuracy, longer inference times may prevent their use in real-time applications. Across various datasets, the authors reported inference times ranging from 1.81 ms to 4.74 ms. A broader perspective was provided by Canziani, Culurciello, and Paszke [196], showing that accuracy and inference time are often in a hyperbolic relationship, where marginal improvements in accuracy can lead to disproportionately large increases in computational time. Canziani, Culurciello, and Paszke [196] also demonstrated that the number of operations in neural networks can be an effective predictor of inference time. These studies collectively highlight the critical importance of understanding the relationship between model size and inference time in the design and deployment of machine learning models, as larger models often result in longer inference times [131], [132], [154], [192], which can be particularly challenging in contexts where computational resources are constrained or real-time predictions are required.

Solutions for transport mode detection can mainly be implemented remote or locally on-device and depending on the choice of deployment strategy, various factors needs to be considered, such as latency, energy consumption and privacy. Centralised algorithms are hardware-agnostic, allowing the deployment of complex models, that would otherwise be infeasible on resource-constrained mobile devices [56], [61]. Wang, Cao, Yu, *et al.* [68] states that remote solutions inherently enhance model security, as models deployed on controlled servers reduce the risk of unauthorised access or leakage. However, as centralisation concentrates sensitive data in a single location, other authors stress that this can also contribute to heightening privacy risks [197]. Moreover, continuous data transmission to centralised servers is resource-intensive for client devices, leading to substantial data transfer volumes that increase operational costs and deplete mobile device battery life [65], [197]–[199]. Remote systems are also subject to increased latency, as data must travel between the client and the server. This can cause delays and bottlenecks during periods of high demand, reducing system efficiency and responsiveness [65], [200], [201]. Finally, maintaining a stable connection to the server can be difficult in urban or underground environments [197], [202].

Local implementation of transport mode detection solutions offers the advantage of operating entirely on the device, eliminating the need for data transmission to a centralised system and relying solely on the device's internal resources [56], [61]. Additionally, it reduces the computational overhead associated with data transmission while enhancing user privacy [56], [68]. By processing data directly on the device, local approaches also significantly reduce latency, enabling immediate feedback to users [61]. Consequently, many researchers advocate for fully local transport mode detection solutions, where data collection, preprocessing, feature extraction, and inference are performed entirely on the device [14], [56], [127]. While local execution is generally preferred due to its advantages in latency, privacy, and network independence [65], it is not without challenges. Machine learning models designed for on-device execution must be simplified to meet the resource constraints of mobile devices. Larger models can pose difficulties in terms of application installation and updates [61], [68]. Although local solutions eliminate the need for network transmissions, which can lower computational costs and energy consumption [65], [197], [198], running complex models on the device can also result in increased energy usage [68].

A crucial consideration in the deployment of machine learning models, particularly on mobile and embedded devices, is power consumption, which directly impacts the practicality and sustainability of these models in real-life applications [60], [196], as energy efficient solutions can prolong the battery lifetime and cut maintenance costs [203]. Battery behaviour is influenced by factors such as temperature and system load [204], meaning that variations in environmental conditions and workload intensity can further impact energy efficiency and device longevity. The total energy consumed by a given embedded computing applications is the sum of the energy required to fetch data from the available memory storage and the energy required to perform the necessary computation in the processor [159]. However, to better assess energy efficiency, total energy consumption should be considered alongside throughput, as power consumption is influenced by computational volume and may be constrained by the device's maximum power capacity [203]. Throughput, in turn, depends on which sensors, features, and sampling frequencies are employed.

As previously mentioned, location-based solutions that utilise GPS incur a significant overhead in terms power consumption [60], [89], [127], [132], [205]. As such, many researchers have shifted their focus towards sensor-based approaches, which utilise significantly less power. However, the use of multiple sensors, especially tri-axial directional sensors can impose elevated power consumption [72], [135], [206]. Moreover, while higher sampling frequencies often improve classification accuracy, it also leads to high energy consumption [57], [89]. One strategy to reduce energy consumption is to train models at a high sampling rate and applying them during inference at a lower sampling rate [89]. High sampling rates are mainly employed to reduce noise, however newer barometer chips support internal hardware smoothing and thus employing the barometer alleviate the need for higher sampling rates [135]. Because of this, Sankaran, Zhu, Guo, *et al.* [135], proposed an approach using only the barometer and compared the power consumption of their solution to that of Google's Activity Recognition. While achieving comparable accuracy, their solution consumed 32 milliwatts (mW) as opposed to Google's Activity Recognition algorithm, which consumed 35 mW. That being said, researchers have achieved even less energy consuming transport mode detection solutions while employing multiple sensors. Lee, Lee, and Lee [127] combined acceleration data with data from the microphone, and achieved a power consumption of 26.1 mW.

Other than the sensor activations themselves, the processing needed to extract features also imposes additional power consumption, and employing fewer features reduces the energy consumption [72], [133]. Taherinavid, Moravvej, Chen, *et al.* [190] identified seven low-power features for transport mode detection based on experimentation and prior research. The process involved combining insights from existing studies and heuristic evaluations to determine which features were both effective for transportation mode classification and suitable for low-power, resource-constrained applications. These seven low-power features consisted of time and frequency domain features of the accelerometer, magnetometer, and gyroscope. However, the energy-consuming process of converting data from the time domain to the frequency domain introduces a trade-off that warrants careful consideration [133]. Ferreira, Zavgorodnii, and Veiga [56] measured the power consumption of their proposed solution on a range of devices, resulting in an average power expenditure of 2.5% per hour. The authors observed a large variability in energy expenditure between the different devices, which could likely be caused by the battery health of the devices, as batteries lose their capacity over time by performing successive charges and discharges [56]. Oplenskedal, Taherkordi, and Herrmann [14] also measured the power consumed of their proposed solution, on a range of devices of different ages. They measured the energy consumption by extracting battery statistics from the operating system of the devices, resulting in an energy expenditure between 0.5 and 25 milliampere-hours (mAh) depending on test scenario and device age. The authors note that accurate transportation mode detection is complex, and usually a high accuracy entails either an unacceptable power consumption and a high computational overhead, or requiring very long data sequences.

2.6 Identified Research Gaps

Despite significant advancements in transport mode detection (TMD), there remain significant challenges that constrain the development of more generalisable, efficient, and applicable solutions.

2.6.1 Limited Dataset Diversity and Representativeness

One of the most pressing concerns lies in the availability and diversity of datasets. Existing publicly available datasets exhibit limitations in terms of the number of participants, available modalities, and device variety, which hinders the ability to generalise findings across diverse real-life conditions. These datasets frequently fail to account for key human factors such as varying device placements or individual movement patterns, which are vital for robust and adaptable TMD models. Furthermore, the lack of datasets incorporating sensor data from iOS devices limits the development of cross-platform solutions, as existing research primarily depends on Android-based data collection. Additionally, region-specific modes of transportation, such as seagoing vessels or niche transport types, are under-represented, further limiting the applicability of existing datasets.

2.6.2 Lack of Standardised Feature Selection and Evaluation Methods

Beyond dataset limitations, challenges persist in the selection, extraction, and evaluation of features for machine learning models. Existing studies employ a wide variety of sensor configurations, ranging from accelerometers and gyroscopes to geomagnetic rotation vectors and barometers, but there is little consensus on the optimal set of features for TMD. Different feature importance techniques, including model-specific and model-agnostic methods, often yield inconsistent rankings of feature relevance, leading many studies to use domain knowledge to manually engineer features from raw sensor data. This lack of robust, standardised approaches for feature selection and importance, results in models that may include redundant or irrelevant features, increasing computational complexity without necessarily improving classification performance. This inconsistency creates difficulties in comparing results and benchmarking performance across studies. It also hinders the development of universally applicable solutions, as models trained on specific sensor combinations may struggle to generalise to different setups or environments.

2.6.3 Accuracy Limitations in Sensor-Based Classification

The accuracy of sensor-based TMD solutions remains an area requiring substantial improvement, particularly when distinguishing between similar transport modes. Although location-based methods leveraging GPS and GIS data often achieve higher classification performance, they suffer from high energy consumption and privacy concerns, making them less suitable for real-time, on-device implementations. Sensor-based approaches offer a privacy-preserving and energy-efficient alternative but frequently underperform in distinguishing between closely related transport modes such as trams and trains, particularly in urban environments where movement patterns and acceleration characteristics overlap. The variability in device placement further exacerbates this issue, as models trained on data from a specific placement (e.g., pocket, backpack, or hand) may not generalise well to other contexts. Hybrid approaches combining location and sensor data have demonstrated improvements in classification accuracy, but they still rely on location-based features, which reintroduce concerns regarding power consumption and privacy.

2.6.4 Energy Efficiency and Privacy

Although outside the primary scope of this thesis, challenges related to energy efficiency and privacy remain critical considerations. Energy efficiency is a key concern for local, on-device processing, as many existing solutions depend on high-frequency sensors such as GPS and accelerometers to achieve high accuracy, leading to significant battery drain and making long-term mobile deployment impractical. Balancing energy efficiency with accurate and granular mode classification remains an ongoing challenge, particularly for resource-constrained devices. Privacy concerns also persist, as many TMD systems rely on sensitive data, such as GPS trajectories and accelerometer readings, which can inadvertently reveal personal information, including travel habits, home and work locations, and lifestyle patterns. These privacy risks are further amplified in centralised architectures, where raw data transmission to remote servers increases vulnerability to unauthorised access and potential breaches. While this research acknowledge current limitations regarding energy efficiency and privacy, and explore potential mitigations, they remain open research challenges in the field.

Addressing the above mentioned challenges requires focused efforts to enhance dataset diversity, establish standardised methodologies, and design solutions that are both platform-agnostic and capable of maintaining high accuracy across an extensive range of transportation modes. Such advancements are essential to push the boundaries of Intelligent Transportation Systems and ensure their practicality in diverse, real-life scenarios.

Chapter 3

Methodology

This chapter positions this thesis, and describes the approach taken in order to investigate the research questions posed in Chapter 1 based on the research gaps identified in Chapter 2. This chapter contains details regarding the data collection phase, data preparation, and algorithmic configuration, including a preliminary investigation into the collected data.

3.1 Research Paradigm

This section outlines the philosophical stance and methodological approach underpinning this thesis. It explains the research paradigm, epistemological position, and how these inform the iterative design and evaluation of artefacts developed in this work.

3.1.1 Philosophical and Methodological Positioning

This thesis is positioned within the Design Science Research (DSR) paradigm, a framework that is uniquely suited to address real-life, applied research challenges through the iterative development, evaluation, and refinement of innovative artefacts [207]. In this research, the primary artefacts developed include a novel dataset for mobility research, high-accuracy machine learning models for transport mode detection, a framework for feature evaluation and reduction, and a platform-agnostic framework for on-device transport mode detection. These artefacts serve practical needs, such as enabling seamless public transportation ticketing and providing accurate travel metrics that can enhance public transport planning and route optimisation for municipalities and other stakeholders.

This research adopts a positivist epistemology, which asserts that knowledge must be developed objectively through empirical observation, measurable data, and hypothesis testing

[208]. The positivist perspective aligns well with the DSR paradigm, as it focuses on quantifiable, empirical evidence derived from systematic data collection and rigorous evaluation. By employing machine learning algorithms and performance metrics such as accuracy, precision, recall, and F1-score, this research grounds its findings in objective and reproducible measurements. The emphasis on quantification and empirical validation ensures that the developed artefacts not only address theoretical challenges but also demonstrate practical utility within real-life settings. Positivism also shapes the evaluation strategies used in this thesis. The reliance on real-life sensor data and statistical methods for feature engineering, model training, and artefact evaluation reflects a commitment to empirical rigour. Furthermore, the iterative nature of DSR allows for continuous refinement of the artefacts, ensuring that they are both scientifically robust and practically relevant. This approach, informed by positivist principles, ensures that the research contributes to the advancement of transport mode detection by producing solutions grounded in objective evidence and systematic inquiry.

The DSR paradigm emphasises constructing artefacts that directly address real-life design tasks faced by practitioners, aiming to provide solutions that are both functional and applicable within organisational settings [209]. This approach is a deliberate departure from the more descriptive aims of natural science, focusing instead on the utility of artefacts and their real-life applications [210]. This research adheres to DSR principles by progressing through successive cycles of relevance, design, and rigour by incorporating real-life data from public transportation to ensure practical applicability, as well as focusing on iterative building and evaluation of artefacts [211]. While the problems and requirements are inspired by industry, this thesis is also grounded in established theoretical frameworks and methodologies relating to intelligent transportation, activity recognition, and machine learning research to ensure a robust theoretical foundation. By integrating methodologies and theories specific to transport mode detection and automated fare collection, this thesis ensures a solid theoretical anchoring, which is essential in DSR for maintaining rigour and advancing the knowledge base. The practical relevance of DSR is addressed through the artefacts' design to meet specific needs in a public transportation context. In this thesis, the frameworks and models are designed to operate effectively in Intelligent Transportation Systems, to facilitate mobility analytics, and for optimising fare management and service delivery. This attention to practical utility is a key component of DSR, where relevance and utility of the artefact in its intended environment are paramount [212].

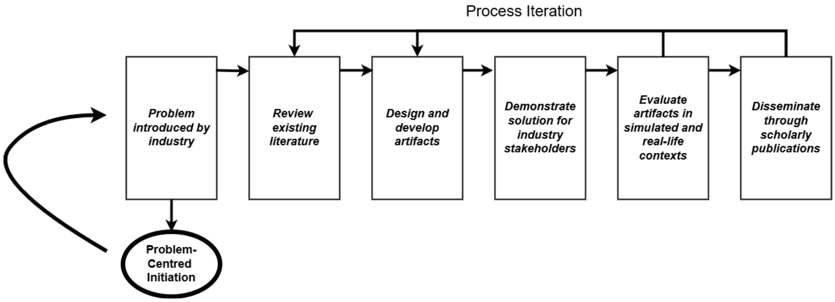


Figure 3.1: DSRM Process Model employed in this research. Inspired by [209].

3.1.2 Design Cycles and Artefact Development

Following the DSR approach, the development of the artefacts presented in this thesis follows a systematic, artefact-centric, iterative process. This thesis was initiated in response to a problem statement identified within the industry. To gain deeper insights into this issue, a comprehensive literature review of automated fare collection solutions and their enabling technologies was conducted (Article I). The findings from this review highlighted key challenges and opportunities, prompting a shift in focus towards transport mode detection. Despite advancements in transport mode detection, prior research has identified several persistent challenges that limit the development of generalisable and efficient solutions. These include the lack of diverse and representative datasets, inconsistencies in feature selection and evaluation methodologies, and the difficulty of achieving accurate, efficient, and platform-agnostic on-device inference. Existing datasets often fail to capture real-world variability in user behaviour, device placement, and transport mode diversity, hindering model generalisability. Moreover, the absence of standardised feature evaluation techniques results in models that may include redundant or irrelevant features, reducing efficiency. Finally, while location-based methods improve accuracy, they introduce privacy concerns and high energy consumption, making them unsuitable for real-time deployment on mobile devices.

These challenges related to predictive accuracy, platform compatibility, feature importance, and computational efficiency directly inform the research questions posed in this thesis, particularly the need for high-accuracy, generalisable models, a systematic framework for feature evaluation and reduction, and the optimisation of on-device inference for real-time applications. Grounded in these challenges, this research progressed into the creation of an initial transport mode detection framework, informed by features based on smartphone sensor data identified from the literature as significant for transport mode detection (Article II). Each subsequent iteration applies a range of processing techniques, feature engineering, and algorithmic approaches to refine the framework, thus progressively enhancing their functionality and accuracy (Article III and Article IV). These iterative refinements are evaluated partly through real-life testing and performance metrics (Article V), such as precision, recall, F1-score, inference speed, and energy efficiency. This iterative process is outlined in Figure 3.1, inspired by Peffers, Tuunanen, Rothenberger, *et al.* [209]. While Figure 3.1 illustrates the overarching methodology employed in this research, the individual design cycles vary in their points of initiation. A design cycle, in this context, refers to the work conducted toward either a specific contribution or a scholarly publication. Specifically, whereas the first design cycle followed a problem-centred initiation, subsequent iterations commenced through different entry points. Given that the overarching research problem had already been established, later design cycles were initiated either through a literature review or by designing and developing new artefacts or refining existing ones based on the findings of the preceding cycle. This iterative process is depicted in Figure 3.2.

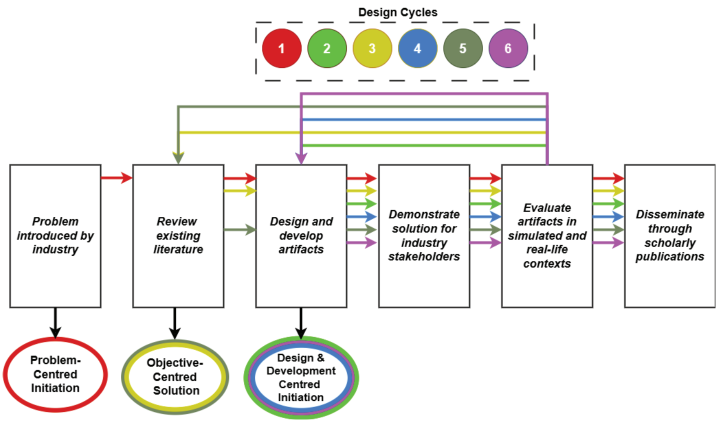


Figure 3.2: DSRM process for each design cycle conducted in this research. Inspired by [209].

The use of iterative cycles, enables this thesis to contribute both utility by delivering functional artefacts, as well as scientific rigour by advancing theoretical understanding in transport mode detection and on-device inference. Evaluation of the artefacts is a central component of DSR, ensuring that each iteration not only meets theoretical expectations but also performs effectively in practice. This thesis adopts a quantitative evaluation strategy. Using a large dataset of real-life sensor data collected from public transport vehicles, each iteration of the different frameworks is rigorously tested and validated against empirical metrics. This quantitative approach provides a reliable measure of performance, ensuring that the frameworks achieve high levels of accuracy and robustness. In DSR, relevance is as critical as rigour, as the created artefacts should address tangible problems within their environment [207]. The frameworks of this thesis are designed with practical utility as a primary objective, offering various results towards improving intelligent transportation and enabling seamless automated fare collection solutions.

3.2 Research Approach

Building upon the DSR paradigm, this section details the specific research approach employed in this thesis, including data collection, feature engineering, and model evaluation. This approach is structured to ensure that each stage of artefact development aligns with the principles of DSR, particularly the iterative build-and-evaluate cycles, which is essential for maintaining both practical relevance and scientific rigour. During the initial stages of the research period, regular meetings were held with industry stakeholders and public transport operators to deepen the understanding of the problem, the domain, and specific requirements. In these discussions, public transport operators emphasised the importance of minimising infrastructure requirements, and expressed a strong preference for solutions that would not require the installation of physical equipment on board vehicles due to concerns regarding scalability and cost. Consequently, the focus shifted from the challenge of in-vehicle presence detection in AFC solutions to transport mode detection in general, as this can be achieved locally without the need for external equipment. This approach not only meets the scalability and cost requirements but can also contribute to in-vehicle presence detection in a more generalisable manner.

3.2.1 Data Collection

To advance the field of intelligent transportation and transport mode detection, a comprehensive and representative dataset was essential. Existing datasets lacked diversity across platforms, devices, sensors, and transportation modes, necessitating the collection of a new, more diverse dataset. Effective data collection is a cornerstone of research in Intelligent Transportation Systems, serving as the foundation for reliable model training and evaluation. To develop a robust dataset, the process must address key principles related to representativeness and accuracy. Representativeness ensures the dataset reflects the diversity of real-life conditions, capturing variations in user demographics, geographic locations, and transportation modes. Accuracy is paramount, requiring data to be precisely labelled and free from noise or inaccuracies that could compromise subsequent analyses. The importance of representativeness and accuracy formed the foundation of the data collection strategy implemented in this thesis, as the lack of representative data and erroneous labelling can significantly impact the performance and reliability of the proposed solutions.

Data was collected by recruiting regular travellers in Oslo and Bodø, Norway, who used a custom mobile application to collect data during their regular travels. A total of 101 participants, including both women and men across all age groups with varied occupations, were recruited through different communication channels in these two cities. Information brochures were distributed to travellers at public transport hubs throughout the two cities, and recruitment efforts extended to festivals and community events organised by the municipalities. Additionally, information brochures were provided to university students in both cities, resulting in a broad and diverse sample of participants for the data collection.

Before the data collection phase commenced, all participants attended a two-hour, in-person training session. This session included information regarding the research project, its main goals, and how the participant's data was processed. This training session also emphasised the importance of accurate data labelling, with clear instructions to delete and refrain from uploading data if they had any uncertainty about its accuracy. During this session, participants were divided into small groups, and accompanied by instructors, conducted several public transport trips to ensure familiarity with the application and proper data recording techniques. Following the training, participants collected data over a one-month period, and were provided a complimentary public transport ticket as an incentive. Support was offered by email throughout the collection period, and any reported incorrectly labelled data uploaded by mistake, was removed from the database. Data collection was conducted using the mobile application during the participants' regular journeys. Figure 3.3 displays the user interface of the developed data collection application. During each trip, participants selected the device's placement location from options including *hand*, *pocket*, and *other*. If the device was held by the participant, they would select hand. If stored in a pocket, they selected pocket. The other category included placements such as in a backpack or purse, or in cases like car or bicycle travel, locations such as a car's centre console or a bike phone mount.

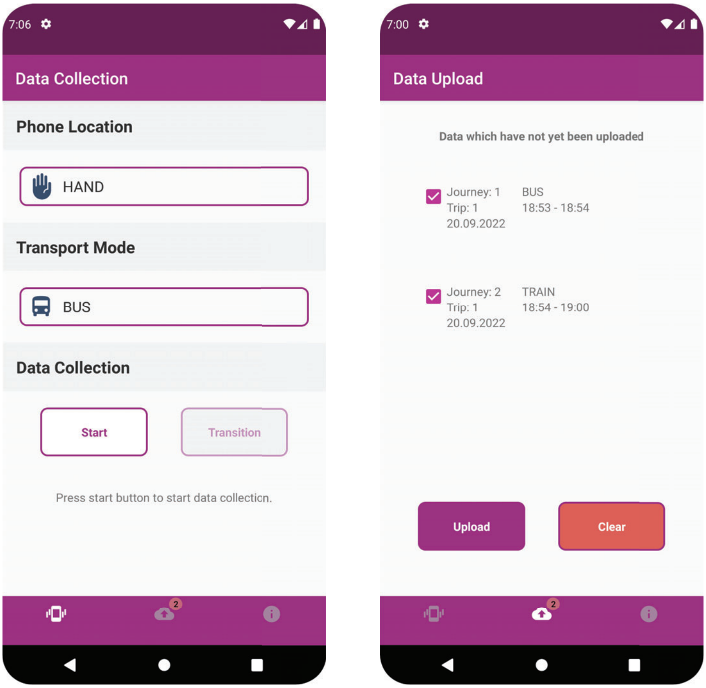


Figure 3.3: User interface of the data collection application. **Left:** Displays the data collection tab of the application. **Right:** Shows the verification and data upload tab of the application.

After selecting the device placement, participants would then choose the mode of transportation in which they were collecting data. The available options were *bus*, *train*, *tram*, *metro*, *e-scooter*, *bicycle*, *boat*, *car*, *inside*, or *outside*. While most of these options are self-explanatory, additional clarification is needed for the terms inside and outside. Participants were instructed to select inside when they were located within a building and *not* inside any vehicle. Conversely, outside was used when participants were outdoors without the use of vehicles such as e-scooters or bicycles. The outside and inside modalities were designed to distinguish between transport and non-transport contexts, enabling comparison and helping to detect dissimilarities across similar transport modes.

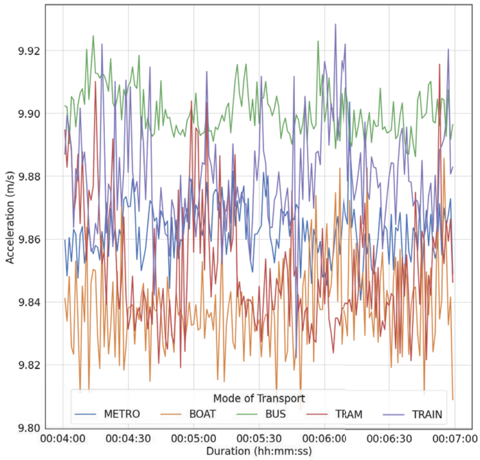
Upon initiation, the application would listen to and capture all available sensor events on the device. This data included readings from both base sensors and composite sensors, as well as any manufacturer-specific sensors present on the device. Base sensors are hardware-based components designed to measure motion or ambient conditions directly, such as accelerometers or magnetometers. Composite sensors, on the other hand, provide data that is derived from processing or fusing multiple base sensor readings, examples of which include orientation and rotation vectors. In addition to sensor data, the application also captured fine-grained location information and activated the device's microphone to record sound. To ensure privacy, only the peak amplitude of the audio was recorded. This was calculated using the following formula:

$$A_{\text{peak}} = \max_{i \in [0, \text{readSize}-1]} |x_i|$$

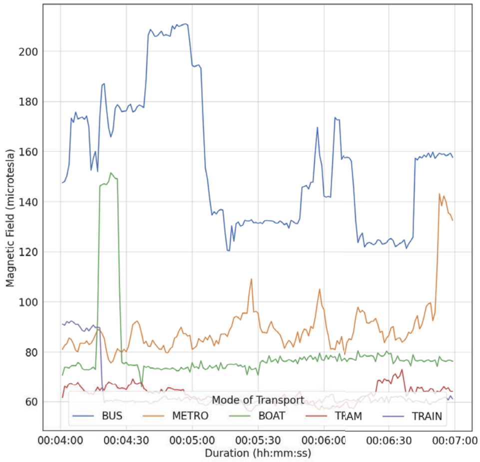
Here A_{peak} represents the peak amplitude, x_i denotes the i-th audio sample in the buffer, and $readSize$ is the total number of samples read from the audio buffer. Data was continuously collected until the participant explicitly stopped the recording. Once data collection was halted, the trip data was displayed in the application's upload tab. In the upload tab, participants could verify that they had selected the correct mode of transportation and that they had stopped data collection at the appropriate time, prior to disembarking. If a participant had forgotten to stop the collection or had selected an incorrect mode, they had the option to delete the data before uploading. After verification, the participant would upload the data to a centralised repository for further processing.

3.2.2 Preliminary Analysis

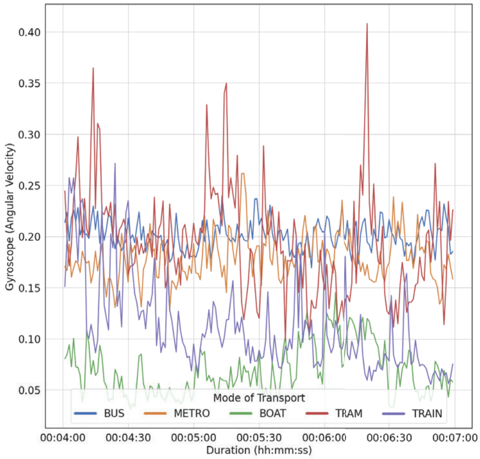
A crucial aspect of machine learning research and development is the thorough analysis and preparation of data. Therefore, a preliminary statistical and time series analysis of the collected dataset was conducted to assess its applicability. This preliminary analysis aimed to assess whether the collected sensor data exhibited sufficient differentiation across transport modes to support accurate classification by machine learning algorithms. Specifically, the analysis sought to identify patterns and variations in sensor readings that are characteristic of different modes of transportation, which would enable the selection of relevant features for model training. Additionally, the analysis aimed to evaluate the suitability of each sensor type (e.g., acceleration, gyroscope, magnetic field, and pressure) in capturing unique signatures associated with specific transport modes, thereby informing decisions on which sensors would contribute most effectively to transport mode detection. A secondary objective was to assess the data quality, particularly with regard to the presence of outliers, as outlier treatment could significantly impact the reliability and performance of the models.



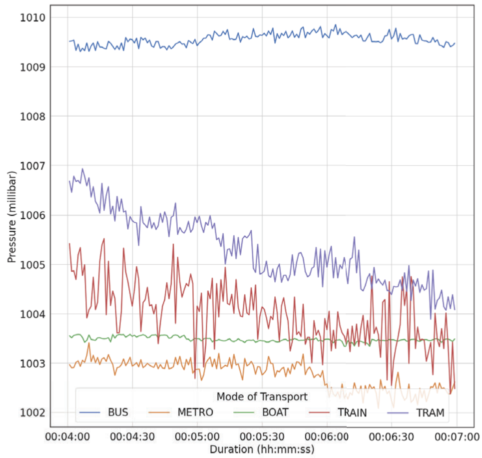
(a) Average acceleration (m/s)



(b) Average magnetic field (microtesla)



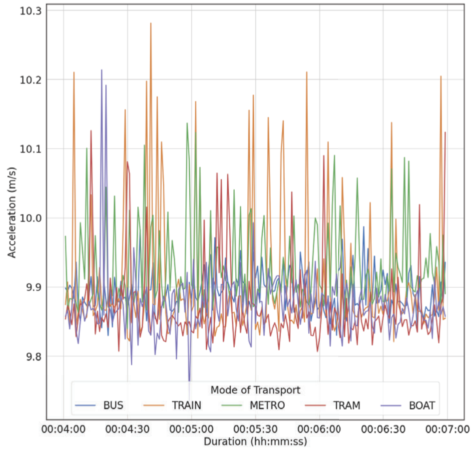
(c) average gyroscope (angular velocity)



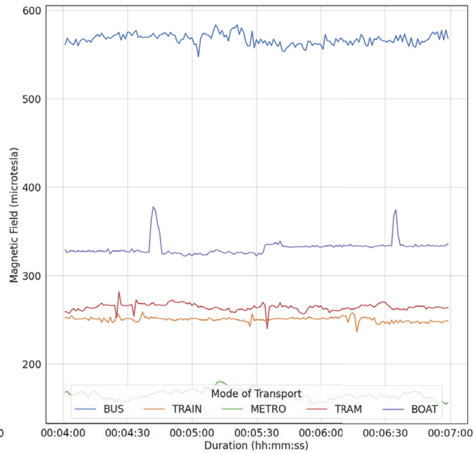
(d) Average pressure (millibar)

Figure 3.4: Average sensor values on Android devices over time per mode of transportation.

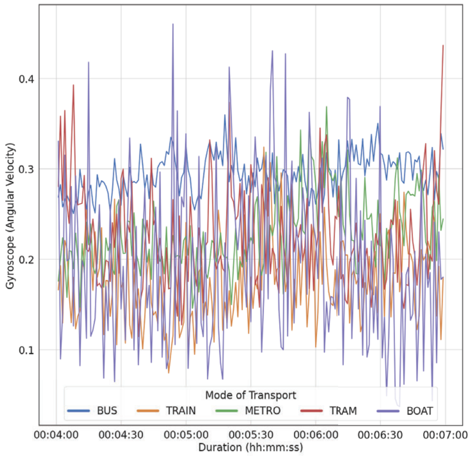
Figures 3.4 and 3.5 display the average sensor values for different transportation modes, aggregated across all participants, across four different sensors collected on Android and iOS devices respectively. For visualisation purposes, only the most prominent public transport modes were analysed, using a small subset of the available sensors. A set of time series plots was generated, capturing the average sensor values over different durations on board five different modes of transportation. The data was aggregated across all participants to provide an average representation of three minutes on public transport.



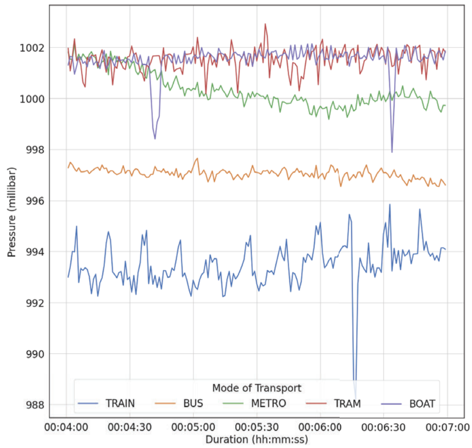
(a) Average acceleration (m/s).



(b) Average magnetic field (microtesla).



(c) Average gyroscope (angular velocity).



(d) Average pressure (millibar).

Figure 3.5: Average sensor values on iOS devices over time per mode of transportation.

Specific patterns were challenging to identify visually for acceleration data, despite its prominence in the literature for Smart Mobility and Intelligent Transportation Systems. The gyroscope exhibited slightly more variation across modes, suggesting that this sensor could reveal detectable patterns for machine learning algorithms. In contrast, magnetic field and pressure data displayed distinct values across different transport modes, indicating clearer potential for mode differentiation. These trends were consistent across Android and iOS devices, although the actual values varied between platforms, likely due to differences in hardware and software. The initial time series analysis across different sensors, transport modes, and time frames, indicated that machine learning algorithms for transport mode detection, based solely on the collected dataset consisting of smartphone sensor data, could be a viable approach. Having assessed the feasibility of this approach, it became necessary to evaluate data quality, particularly concerning outliers. Various techniques are available for identifying outliers. Two widely used methods are based on either standard deviation or the interquartile range (IQR). Standard deviation is less robust and may be misleading for non-normally distributed datasets. The collected data was quite skewed, suggesting that a more robust approach, such as using the IQR, would be more suitable for identifying outliers. The interquartile range (IQR) is defined as:

$$\mathrm{IQR} = Q_3 - Q_1$$

where Q_1 is the first quartile (25th percentile) and Q_3 is the third quartile (75th percentile). The bounds for identifying outliers are given by:

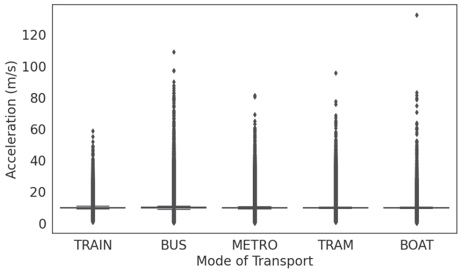
$$\text{Lower Bound} = Q_1 - 1.5 \times \mathrm{IQR}$$

$$\text{Upper Bound} = Q_3 + 1.5 \times \mathrm{IQR}$$

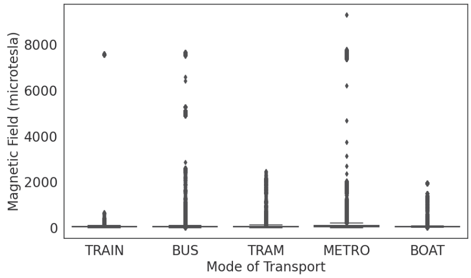
A data point x is considered an outlier if:

$$x < Q_1 - 1.5 \times \mathrm{IQR} \quad \text{or} \quad x > Q_3 + 1.5 \times \mathrm{IQR}$$

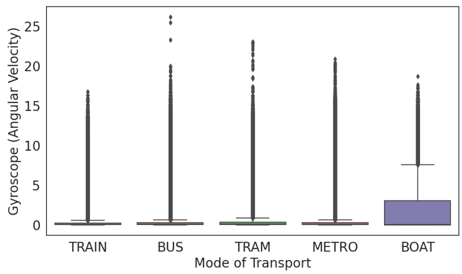
Figures 3.6 and 3.7 illustrates the distribution of sensor values for each transport mode, with outliers highlighted using the interquartile range (IQR) method. Across all four sensors, for each transport mode, and on both Android and iOS platforms, a substantial number of outliers were identified. Due to the significant amount of outliers, removing outliers could lead to sensor values that are overly similar across different transportation modes, potentially diminishing the variability necessary for effective classification. In the literature, there is no clear consensus on whether removing outliers is beneficial for transport mode detection, as the impact of outliers on model performance varies with the specific problem and dataset. In some cases, outliers may capture unique variations in sensor behaviour that correspond to specific, less frequent occurrences within a transport mode (e.g., sudden accelerations in a bus due to braking or uneven road surfaces on a tram line). It is difficult to differentiate noise from outliers without the help of an expert [172] and retaining these data points may therefore provide additional, valuable information that can enhance the model's ability to differentiate between subtle variations in transport modes, thereby potentially improving classification accuracy.



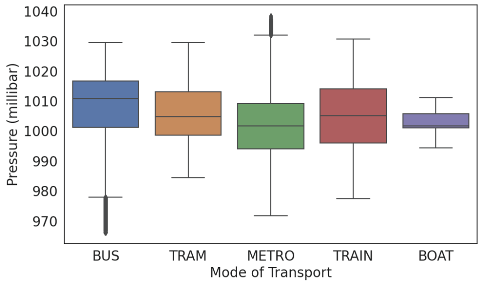
(a) Acceleration (m/s)



(b) Magnetic field (microtesla)

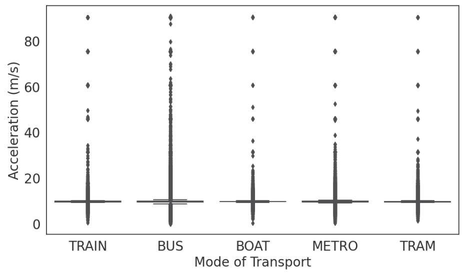


(c) Gyroscope (angular velocity)

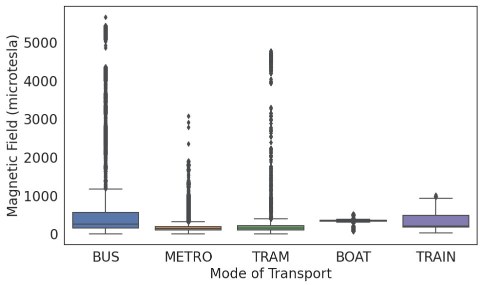


(d) Pressure (millibar)

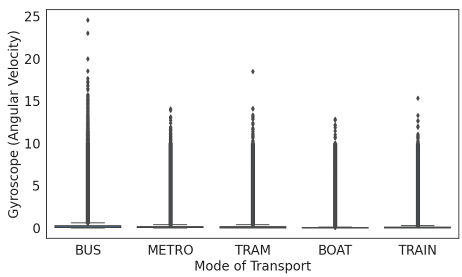
Figure 3.6: Distribution of sensor data across transport modes on Android devices.



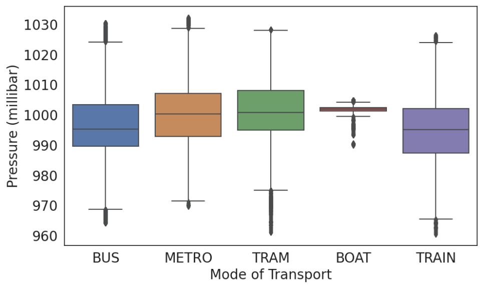
(a) Acceleration (m/s)



(b) Magnetic field (microtesla)



(c) Gyroscope (angular velocity)



(d) Pressure (millibar)

Figure 3.7: Distribution of sensor data across transport modes on iOS devices.

Given this ambiguity, along with the potential for outliers to reflect meaningful contextual information rather than noise, outliers were retained in the dataset. This approach ensures that the model can access the full range of sensor data variability, thereby capturing distinct and possibly informative patterns associated with each transport mode. By examining the dataset's potential for transport mode differentiation and evaluating data quality, this analysis provides a foundational understanding that guides the artefact's development.

3.2.3 Preprocessing and Feature Importance

Based on the insights gained from the preliminary analysis, feature engineering was undertaken to transform the raw sensor data into informative features that could effectively represent each transport mode. In line with DSR principles, the feature engineering process was guided by practical relevance and the unique characteristics identified within the dataset. By carefully selecting and creating features that captured the distinct patterns observed across sensor types, the goal was to enhance the model's ability to differentiate between transport modes with high accuracy. Throughout this research period, a variety of preprocessing and feature selection techniques have been explored. From the literature, a popular approach to handle large amounts of temporal sensor data is to segment and aggregate the data. In this context there are mainly two aspects that are important when it comes to preprocessing: the function used for aggregation and the way data is segmented into windows.

While the research community seems to agree on the approach using aggregation and segmentation, there are very large discrepancies in terms of window size and aggregation functions. The length of the sliding window is usually between one and 60 seconds, however longer windows can also be reasonable, although there is a trade-off between granularity and less resource-consuming inferences and higher window density [73]. The literature does not present any conclusions as to which configuration is optimal for transport mode detection in terms of segmentation, and as such various configurations were explored during this research period. Window lengths of 10, 20 and 30 seconds with a 5-second overlap (step size) were initially investigated (Article II and Article III) where a configuration of 10-second windows with a 5-second overlap was chosen, since a smaller window reduces the required pre-inference data collection and negligible differences were observed when increasing the window size. However, later experimentation revealed that increasing the size of the window when simultaneously decreasing the overlap significantly impacted the results (Article V). As such, segments with window lengths up to 60 seconds were explored with overlap from 5 to 0.1 seconds. A stagnation of the accuracy was observed around 1-second overlap. Using an overlap of 1 second while increasing the window size, it was observed that near-optimal accuracy had already been reached at 15 seconds. Since the pre-inference data collection on device is directly influenced by the size of the window, the best configuration in the context of on-device transport mode detection was found to be 15-second windows with a 1-second overlap.

It is important to note that assessment of the segmentation parameters was based on the accuracy of a model trained with data segmented using 15-second windows with a 1-second overlap. As such, the results also relied on other aspects of the data, such as features and normalisation techniques used, and as the research progressed these aspects naturally changed. As a result, early segmentation experiments cannot be directly compared with later assessments. Another essential aspect of data preprocessing for transport mode detection is the selection of aggregation functions. Given that the data is segmented into windows, it is necessary to aggregate sensor readings within each segment.

Previous studies have employed a variety of aggregation functions, with considerable variation and no established consensus on which functions are most suitable for mobile sensor data in the context of transport mode detection. Similarly, significant discrepancies exist regarding the use of inertial and ambient sensors as features in machine learning models for transport mode detection. Many of the ambient and inertial sensors available in modern smartphones capture directional data across the device's three perpendicular axes (x, y, and z), reflecting movement or environmental values along these dimensions. To mitigate directional dependency and provide a more uniform measure, it is possible to compute the magnitude of these three axes. The magnitude can be given as:

$$magnitude = \sqrt{(x^2 + y^2 + z^2)}$$

This transformation removes axis-specific variability, enabling the model to focus on overall intensity rather than direction, which can be particularly useful for transport mode detection. In the literature, some studies choose to use directional values to capture specific axis-based movements, while others calculate the magnitude to achieve a more generalised measure of movement. During this research period, both approaches have been assessed during different iterations.

In order to identify the optimal combination of sensors and aggregation functions one can employ different techniques to rank the importance of each feature. Some algorithms have built-in feature importance functionality such as XGBoost and Random Forest. However, which features are deemed important in the context of decision trees, such as XGBoost and Random Forest, are not necessarily equally important in the context of neural networks. Some more generic approaches exists, and the most prominent are Permutation Importance (PI) [213], Shapley Additive Explanations (SHAP) [214], Mutual Information (MI) [215], Analysis of Variance (ANOVA) f-test [216], and Classification and Regression Trees (CART) [217].

Permutation importance is a technique used in machine learning to measure the importance of features in a predictive model. It works by randomly shuffling the values of a single feature and observing the effect on the model's performance. The decrease in performance after shuffling indicates the importance of the given feature. By calculating the average decrease in performance across multiple shuffles, the features can be ranked based on their importance in the model. Mutual information is a measure of the dependency between two variables and quantifies how much information a feature contains about the target variable by measuring their statistical dependence. Analysis of variance f-test is a statistical test that assesses whether the means of two or more groups are significantly different from each other. It calculates an F-score, which can be used to rank the features. SHAP, on the other hand, is a game theoretic approach designed to explain the output of any machine learning model. Finally, the classification and regression trees feature importance is a measure of how much a specific feature contributes to the decision-making process when building a decision tree. The different feature importance approaches all have their strengths and weaknesses and when used to gauge the importance on the same dataset, they all produce radically different results. During the different iterations of this thesis, various feature importance methods were utilised, such as extracting the intersection across multiple feature importance techniques (Article II and Article IV) and iterative feature removal (Article III).

Since participants collected data during their regular travels, the dataset underpinning this thesis exhibited significant class imbalance, with certain transport modes represented far more frequently than others. This imbalance resulted in the dataset being divided into majority and minority classes. A substantial disparity in class representation poses challenges for classification models, as the classifier may become biased toward majority classes, leading to reduced predictive accuracy for under-represented modes [169]. To address this issue, the dataset required rebalancing. Given its broad acceptance in the research community, Synthetic Minority Over-sampling Technique (SMOTE) [170] was applied to resample the minority classes in this dataset. This approach was selected as it preserves all original information in the dataset while synthetically generating new samples in minority classes to improve model balance. Following SMOTE application, a statistical comparison of the dataset was conducted before and after resampling to ensure that the algorithm had not significantly altered the dataset's overall distribution or introduced artificial bias. The results of this comparison is shown in Table 3.1.

Table 3.1: **Statistical comparison between the original data (O) and the synthetic data (S)**: A=Accelerometer, M=Magnetic Field, G=Gyroscope, P=Pressure, X,Y,Z=Axial direction, N=Magnitude.

	Mean		Std		Skewness		Kurtosis	
	O	S	O	S	O	S	O	S
AX	0.356	0.304	2.896	2.802	0.496	0.638	3.777	4.228
AY	0.616	0.709	3.787	3.764	-0.223	-0.156	0.853	0.854
AZ	1.397	1.421	4.791	4.723	0.003	0.001	-0.337	-0.298
AN	5.706	5.582	4.498	4.492	-0.070	-0.022	-1.949	-1.958
MX	59.114	22.240	560.831	431.821	6.632	9.096	51.954	96.707
MY	13.823	11.211	338.520	324.417	8.884	11.080	120.429	164.987
MZ	-105.633	-74.987	462.705	396.437	-3.108	1.109	52.410	73.165
MN	297.363	242.473	755.598	630.708	5.972	8.161	40.245	76.656
GX	0.003	0.002	0.034	0.031	2.189	1.612	115.822	111.347
GY	0.001	0.002	0.049	0.044	-2.829	-3.416	233.897	264.774
GZ	-0.001	-0.001	0.051	0.045	0.548	0.085	116.455	117.496
GN	0.314	0.278	0.426	0.399	3.412	3.703	15.881	18.968
P	573.948	562.060	452.807	452.778	-0.091	-0.041	-1.991	-1.997

While most changes in the dataset after applying SMOTE were modest, significant shifts were observed in the magnetic field sensor data. The mean, spread, variability, and distribution of magnetic field readings (MX, MY, MZ, MN) changed notably, as reflected in increased skewness and kurtosis values. Although these shifts in magnetic field data were pronounced, models trained on the synthetically balanced dataset demonstrated significantly improved performance during experimentation. Consequently, the alterations observed in the magnetic field sensor data were deemed acceptable, as they contributed to a more effective model without compromising the dataset's overall integrity.

Finally, various scaling techniques for normalisation or standardisation were explored. Scaling is essential for machine learning algorithms that rely on distance-based metrics, such as the Multilayer Perceptron (MLP), which uses gradient descent methods for training. Without scaling, raw feature values may cause gradients to be excessively large or small, potentially destabilising the training process. By applying scaling, this issue is mitigated, resulting in more stable and efficient training. A range of scaling techniques were considered, including MinMaxScaler, StandardScaler, RobustScaler, and QuantileTransform [218]. Some scaling techniques, such as StandardScaler, are sensitive to outliers. However, given that outliers were retained in the dataset, it was necessary to select scaling methods robust to their presence. Consequently, RobustScaler, QuantileTransform, and MinMaxScaler were evaluated.

RobustScaler and QuantileTransform are particularly suited to datasets with outliers, with RobustScaler centering the data by the median and scaling according to the interquartile range, and QuantileTransform reshaping feature distributions to follow a uniform distribution. MinMaxScaler, while not able to reduce the effects of outliers, was also evaluated due to its ability to linearly scale features within a fixed range. RobustScaler, QuantileTransform, and MinMaxScaler were evaluated and tested on both Android and iOS devices in a real-life setting, onboard a variety of public transportation vehicles in order to gauge which technique was more suitable for improving models for transport mode detection. During the real-life test, scaling using the QuantileTransformer yielded the poorest performance, with over 90% of classifications incorrectly labelled as "other," despite the majority of classifications occurring on bus and metro. The models scaled with MinMaxScaler and RobustScaler performed well and almost equivalent to each other. The MinMaxScaler is one of the most widely recognised approaches for normalisation and have outperformed other scaling techniques in previous works [218]. Consequently, MinMaxScaler was selected for normalising the data, which transforms the data by rescaling each feature to a range between 0 and 1.

In line with the iterative approach central to DSR, the preprocessing and feature engineering steps were revisited and refined through multiple cycles. Each iteration involved preparing the dataset, developing an initial model, and assessing its performance, with insights from one cycle informing improvements in the next. This process of iterative refinement allowed artefacts to evolve together, enhancing their alignment with the practical requirements of transport mode detection.

3.2.4 Algorithms and Hyperparameters

In developing a framework for local transport mode detection (TMD), a diverse range of algorithms were implemented and rigorously evaluated, including deep feedforward neural networks (DFNN), recurrent neural networks (RNN), convolutional neural networks (CNN), as well as hybrid architectures combining both RNN and CNN. Additionally, traditional machine learning algorithms were assessed, including decision trees, support vector machines (SVM), and k-nearest neighbours (KNN), along with more advanced approaches like gradient boosted trees (XGBoost) and ensemble methods such as Random Forest. All of these algorithms were explored and refined over multiple development cycles. Experimentation revealed that certain algorithms consistently outperformed others. Specifically, DFNNs and RNNs demonstrated strong performance relative to SVMs and KNNs. Similarly, gradient-boosted trees and ensemble tree methods also yielded promising results. Consequently, these initial findings guided subsequent iterations, emphasising models with multilayer perceptrons (MLP, a type of DFNN), long short-term memory networks (LSTM, a type of RNN), convolutional neural network (CNN), as well as tree-based algorithms such as XGBoost and Random Forest.

While much of the model training process focuses on data processing and feature engineering, as discussed in prior sections, selecting suitable hyperparameters for each algorithm is also crucial to performance. Hyperparameters are model-specific configurations that are set before training begins and control aspects of the learning process, such as the learning rate, regularisation strength, or maximum depth in decision trees. For neural networks, hyperparameters include the number of layers (stages of transformation the data undergoes) and nodes (individual units within each layer that process data). The configuration of layers and nodes is essential to model complexity, as additional layers enable deeper feature extraction, while more nodes within layers can capture finer data details. However, finding the optimal combination of these hyperparameters requires careful tuning to balance model complexity and computational efficiency.

Throughout the iterative cycles of this thesis, MLP LSTM, CNN, XGBoost and Random Forest have all undergone continuous refinement and optimisation in terms of configuration and performance. Determining the optimal configuration for machine learning algorithms is particularly challenging, as the effectiveness of a given configuration only becomes apparent after the training phase, which is often time-consuming and resource-intensive. While many researchers and practitioners rely on experimentation to identify effective configurations, systematic frameworks such as grid search and random search are available to optimise this process. Grid search involves exhaustively testing all possible combinations of specified hyperparameters, providing a comprehensive, though often time-intensive, approach to configuration. In contrast, random search randomly selects combinations within a defined parameter space, offering a more efficient alternative by exploring a broader range of configurations without testing each one exhaustively. Both grid search and random search were explored with varying levels of success.

Initial experimentation provided a foundational understanding of parameter ranges suited to this problem, and subsequent attempts with random search did not yield performance improvements beyond what had been achieved through manual tuning. Consequently, random search was set aside in favour of grid search in later iterations, leveraging insights gained from prior experimentation and the literature [219] to refine parameters more systematically.

Table 3.2: Grid search hyperparameters.

Optimisation		
Parameter	**Values**	**Best**
Learning Rate	0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009	0.0004
Activation	relu	relu
optimiser	adam, rmsprop	adam
Initializer	glorot_uniform, he_normal	glorot_uniform
Architectural		
Parameter	**Values**	**Best**
# Layers	1 to 15	14
# Nodes	100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000	800
Android accuracy		99.3 %
iOS accuracy		99.6 %

In later iterations, grid search was employed to surpass the previously achieved performance in the MLP-based approach. Two distinct parameter sets, optimisation and architectural, were defined to address different aspects of the model configuration. By dividing these parameters into separate processes, a broader range of configurations could be tested with reduced resource and time demands. A threshold of 1,000 epochs was established, along with an early stopping mechanism, which halted training after five consecutive epochs without improvement in accuracy.

During the work presented in Article V, the objective was to evaluate a platform-agnostic model for transport mode detection, necessitating assessment on distinct datasets from Android and iOS platforms. To identify an optimal architecture across both platforms, it was essential to find the intersection of configurations yielding the highest accuracy on both datasets. Table 3.2 provides an overview of the parameters and values explored, highlighting the best-performing configuration in each category and the highest accuracy achieved on both holdout sets.

Initially, a third parameter set for regularisation was included. However, since accuracy was the primary success metric, configurations with regularisation were initially suboptimal, as regularisation tends to reduce accuracy while enhancing model robustness and generalisability. To further assess model robustness, the best configuration from the grid search on real devices was deployed and assessed, onboard public transport vehicles. Despite high performance on the holdout dataset, the model exhibited suboptimal accuracy in real-life conditions, suggesting overfitting. Overfitting occurs when a model captures noise or irrelevant details in the training data, impairing its generalisation to unseen data.

To address this, various regularisation techniques were tested, including L1, L2, and combined L1L2 regularisation, alongside dropout layers. L1 regularisation encourages sparsity by penalising the absolute value of model weights, driving some coefficients to zero, whereas L2 regularisation adds the squared weight values to the loss function, reducing the impact of individual features without zeroing coefficients. L1L2 regularisation combines both techniques, balancing sparsity and smaller weight sizes. Implementing regularisation techniques requires setting a lambda value, which determines the regularisation strength, where larger values enforce stronger regularisation. After testing values from 0.00001 to 0.2, it was observed that values above 0.01 impaired learning, as such an upper limit of 0.01 was set. Additionally, dropout layers were applied between hidden layers to prevent overfitting. Dropout layers randomly deactivate neurons during training, promoting the development of robust feature representations. Consistent with prior research [219], a dropout rate of 50% was found be the best configuration of dropout layers to mitigate overfitting in the final model configuration used in Article V.

3.2.5 Evaluation

Throughout the iterative development cycles, trained models were evaluated in both simulated and real-life settings to ensure robust performance. This section details the evaluation methodologies, including the metrics, testing environments, and experimental setups used to validate model accuracy, robustness, and adaptability across diverse transport scenarios. While all models and artifacts were evaluated on unseen real-life data, only a subset was implemented and tested in real-world contexts.

The primary evaluation metric employed in this research was accuracy, which measures the proportion of correct predictions out of the total number of predictions. To further understand classification performance, confusion matrices were utilised, which reveal the distribution of correct and incorrect classifications across each class. The F1-score was also an important metric, combining precision and recall to give a balanced view of the model's performance in handling both positive and negative instances. Precision measured the proportion of correctly identified positive instances among all predicted positives, while recall assessed the model's ability to capture all actual positive instances.

In addition, training time and inference time were evaluated to quantify the impact of dimensionality reduction on model efficiency. Training time was measured by starting a timer at the initiation of model training and stopping it upon completion. Similarly, inference time was measured by initiating a timer just before inputting the preprocessed data into the model and stopping it immediately upon obtaining the model's output. Several models were also deployed on real devices and evaluated in real-life settings onboard public transport vehicles. The real-life implementations included multiple configurations of MLPs and LSTMs, enabling a comparative assessment of model performance under actual operating conditions.

Table 3.3: Overview of devices used for real-life evaluation.

Device	**OS**	**Version**	**Battery Capacity (mAh)**
Sony Xperia 1 (J9110)	Android	11 (30)	3330
Pixel 7a	Android	14 (34)	4385
Samsung Galaxy S21 FE (SM-G990B)	Android	14 (34)	4500
Samsung Galaxy S22 Ultra (SM-S908B)	Android	14 (34)	5000
Samsung Galaxy S23 (SM-S911B)	Android	14 (34)	3900
iPhone 8	iOS	16.7.10	1821
iPhone 13	iOS	17.2.1	3240

Additionally, a configuration of XGBoost was also evaluated on new data, not stemming from the original dataset. It is important to note that not all assessments were conducted with systematic, rigorous evaluation methods. Over the course of this research, hundreds of models were developed, making it impractical to systematically implement and evaluate each one under real-life conditions, as implementation and testing onboard public transport vehicles is highly time-intensive. As a result, many real-life assessments were conducted on a relatively small number of trips and devices, to quickly verify or invalidate results from evaluations performed on unseen data collected during the data collection phase. The comprehensive and systematic assessments were conducted using the devices present in Table 3.3, onboard a variety of public transportation vehicles.

When evaluating models on real devices, factors beyond classification accuracy and inference time become significant. Mobile devices are resource-constrained, particularly in terms of battery life, making it essential to assess the energy impact of model deployment. For Android devices, the energy consumption was measured by extracting energy usage statistics from the operating system using the Android Debug Bridge (ADB). Although the data provides estimations rather than precise measurements, it offers a reliable indication of whether the model imposes an excessive energy demand. For iOS devices, extracting energy usage data after real-life assessments proved challenging. Given that the model configuration was identical across both Android and iOS, and the goal was simply to gauge energy consumption, energy usage was measured exclusively on Android devices.

Chapter 4

Results

Grounded in an iterative process of assessments conducted across both simulated and real-life environments, this chapter details the outcomes of the artefact development central to this research. Each artefact presented in this thesis has been shaped and refined through multiple design cycles, ensuring both rigour and practical applicability. These cycles have informed the development of key contributions, which include: (1) NOR-TMD, a curated dataset designed to advance mobility research, (2) high-accuracy transport mode detection models, (3) EFR-TMD, a generalised feature-ranking framework for identifying critical features across diverse algorithms for transport mode detection, and (4) a platform-agnostic framework for local transport mode detection. In line with the principles of Design Science Research (DSR), these contributions reflect a systematic, iterative process of development and evaluation, leveraging insights gained at each stage to enhance the effectiveness and relevance of the resulting artefacts. To clarify how each article contributes to the overall research objectives and design process, the following section systematically maps the included articles to the key research questions, contributions, and design cycles underpinning this thesis.

4.1 Mapping Articles with Research Questions, Contributions and Design Cycles

The results of this thesis stem from a series of articles, each presenting distinct contributions that collectively address the overarching research questions. The experiments and developments described within these articles were organised following the DSR paradigm, structured as iterative design cycles. This section details how each article corresponds to specific design cycles, key contributions, and research questions, providing a clear and structured overview of how the individual components of this work integrate to advance the overall research goals.

4.1.1 The Link Between Articles and Research Questions

Each research question in this thesis is addressed through the combined findings and developments presented across multiple articles. Rather than a direct one-to-one mapping, the research questions are answered by the integrated outcomes of several studies, reflecting the iterative and comprehensive nature of the DSR paradigm applied in this thesis. Table 4.1 summarises how the collective contributions of the articles align with and support the investigation of each research question. It is important to note that while Articles II–V directly address the core research questions, Article I serves a foundational role by providing a comprehensive survey of the automated fare collection landscape. Article I helped shape the problem context and motivated the subsequent research focus but does not contribute directly to answering the research questions.

Table 4.1: Mapping of Articles to Research Questions

Research Questions	**Article**
RQ0: How can efficient on-device, platform-agnostic transport mode detection be achieved on mobile devices?	II-V
RQ1: How can machine learning models achieve high accuracy in transport mode detection across diverse transport modes, ensuring generalisability in real-life applications?	II-III
RQ2: How can a standardised framework for feature evaluation and reduction systematically identify relevant features, ensuring consistency and enabling reliable feature reduction across machine learning models?	II-V
RQ3: How can transport mode detection models be optimised for real-time, low-latency inference while maintaining computational efficiency on mobile devices?	IV-V

Together, these articles not only provide comprehensive answers to the research questions but also lay the foundation for the key contributions of this thesis. The following section maps the relationship between the articles and the major contributions, highlighting how each piece of work advances the overall research objectives.

4.1.2 The Link Between Articles and Contributions

The key contributions of this thesis, initially outlined in the introduction, are each grounded in specific articles. These contributions include the creation of a comprehensive dataset for transport mode detection and the development of accurate and generalisable classification models. In addition, a framework for feature selection and dimensionality reduction is introduced and evaluated, along with a lightweight, platform-agnostic framework for on-device transport mode detection. Article IV presents the NOR-TMD dataset, a comprehensive, cross-platform dataset collected in real-world settings. Article II and Article III focus on achieving accurate and generalisable transport mode classification using machine learning, leveraging this dataset. Article IV also introduces EFR-TMD, a model-agnostic framework for ensemble-based feature evaluation and reduction. Finally, Article V contributes a lightweight, platform-agnostic deployment framework for on-device transport mode detection. Table 4.2 summarises how each contribution is linked to the corresponding article.

Table 4.2: Mapping of Articles to Contributions

Contribution	**Article**
NOR-TMD: A Comprehensive Dataset for Transport Mode Detection	IV
Accurate and Generalisable Transport Mode Classification	II-III
EFR-TMD: A Framework for Feature Evaluation and Reduction	IV
Lightweight, Platform-Agnostic On-Device Framework	V

4.1.3 The Link Between Articles and Design Cycles

The five articles collectively follow an iterative design cycle, each contributing to different stages of the research process. Article I provides foundational background and motivation. Article II and Article III focus on developing and improving machine learning models for accurate and generalisable transport mode detection. Article IV introduces the EFR-TMD framework for systematic feature evaluation and reduction, addressing model efficiency. Article V validates the optimised models through real-world, on-device deployment. This progression reflects a coherent research trajectory from problem identification to practical implementation, tightly linking the articles to the research questions.

Table 4.3: Mapping of Articles to Design Cycles

Design Cycle Description	**Article**
1: Survey of Automated Fare Collection Solutions in Public Transportation	I
2: NOR-TMD	IV
3: Multilayer Perceptron	II
4: Extreme Gradient Boosting	III
5: Ensemble Feature Ranking Framework	II, IV
6: Platform-Agnostic Framework for Local Transport Mode Detection	V

Table 4.3 exhibits the relationship between design cycle and article. This mapping provides a clear overview of how each design cycle and its related publications contribute to addressing the core research questions and producing the thesis contributions. The subsequent sections describe the design cycles in detail, highlighting methods, results, and their significance.

4.2 Design Cycle 1: Survey of Automated Fare Collection Solutions in Public Transportation

The first design cycle focused on conducting a comprehensive review of automated fare collection (AFC) systems and enabling technologies in public transportation (Article I). Referring to Figure 3.2, this cycle was initially guided by a problem identified in the industry and subsequently advanced through all relevant activities. This foundational effort sought to identify current practices, emerging trends, and critical gaps in existing solutions, laying the groundwork for the subsequent development of innovative artefacts. The review analysed a broad spectrum of literature, technological implementations, and case studies, offering insights into the state of AFC systems and their interplay with IoT, predictive behaviour modelling, sensor analytics, and machine learning. This review revealed that while many AFC systems are in place, they often rely on traditional check-in/check-out (CICO) mechanisms, which require active user interaction. AFC depend heavily on hardware like NFC or RFID-enabled cards, QR code scanners, and mobile apps, which while effective in specific contexts, impose cognitive and operational burdens on users. Furthermore, CICO approaches generate limited data points, offering only transactional records without deeper insights into passenger behaviour.

Limitations related to user interaction and hardware dependency underscore the need for a paradigm shift toward fully automated AFC systems, such as Be-in/Be-out (BIBO) frameworks, which leverage passive sensor data to eliminate the need for user intervention entirely. The review also highlighted that most AFC solutions inadequately address the intricacies of transport mode detection, a crucial requirement for implementing automated systems. Current approaches often fail to seamlessly integrate multi-modal transportation, complicating fare calculations and user experiences. Technologies such as GPS, BLE, accelerometers, and gyroscopes were identified as potential enablers of in-vehicle presence and transport mode detection. However, their application remains fragmented, with significant challenges in achieving precision, energy efficiency, and privacy preservation.

Through discussions with public transport operators in Norway, it became evident that achieving fully autonomous AFC systems without the need for additional hardware is a critical requirement. Given that the majority of the reviewed literature relies on supplementary equipment, this research shifted its focus towards transport mode detection, as it can be implemented without external devices. Furthermore, accurate classification of travel modes serves as a fundamental prerequisite for ticket issuance, thereby facilitating the development of fully autonomous AFC systems.

While this design cycle and the corresponding Article I did not result in one of the primary contributions of the thesis, it played a foundational role. Through a structured review of existing Automated Fare Collection (AFC) systems and enabling technologies, this cycle established the theoretical grounding for the research and helped scope the problem space. Insights from the literature and stakeholder discussions redirected the focus toward transport mode detection as a viable pathway toward fully autonomous fare collection systems without reliance on external hardware. In this way, the cycle contributed to the formulation of the overarching research agenda and motivated the development of subsequent artefacts addressing transport mode detection using inertial and ambient smartphone sensors.

4.3 Design Cycle 2: NOR-TMD

Building on the findings from the previous design cycle, it became evident that a large and diverse dataset was essential to addressing the challenge identified by industry stakeholders. Consequently, this cycle was structured around this objective, beginning with a review of existing literature on dataset requirements and publicly available datasets.

This review revealed a significant lack of publicly accessible datasets, reinforcing the necessity of collecting a large dataset to support the research objectives. This resulted in the NOR-TMD dataset, collected over a one-month period by 101 regular travellers across two Norwegian cities. This dataset is a significant contribution and addresses limitations observed in existing publicly available datasets [72], [123], [183], [184], particularly in terms of devices, participants, sensor variety, and representation of transportation modes. A key strength of this dataset is diversity, as it also covers sensor data collected from both Android and iOS devices. The dataset comprises a total of 609.68 hours of sensor data, covering ten distinct transportation modes and three device placement categories, as can be seen in Table 4.4.

Table 4.4: The number of hours of data collected within each modality.

Mode	Hand	Pocket	Other	Total
BUS	204.39h	65.06h	3.89h	273.34h
METRO	74.14h	26.34h	1.73h	102.21h
BOAT	2.41h	48.24h	25.78h	76.43h
OUTSIDE	14.76h	22.53h	1.15h	38.45h
TRAM	26.93h	6.08h	3.02h	36.03h
TRAIN	19.08h	5.16h	5.2h	29.44h
BICYCLE	0.43h	15.9h	5.52h	21.85h
CAR	3.65h	3.13h	14.27h	21.05h
INSIDE	1.13h	0.85h	5.61h	7.6h
E-SCOOTER	0h	3.28h	0h	3.28h
Total	346.92h	196.57h	66.17h	609.68h

The dataset consists of data from a large variety of devices, including 11 different manufacturers and 57 unique models. The amount of data collected through each device manufacturer is presented in Figure 4.1. Data was collected from 14 different sensors across both Android and iOS devices, as detailed in Figure 4.2. Additionally, data from the devices' built-in activity recognition was recorded and included in the dataset. However, since this feature operates at a very low frequency, only providing updates when a context change is detected, it has been excluded from Figure 4.2 for clarity and readability.

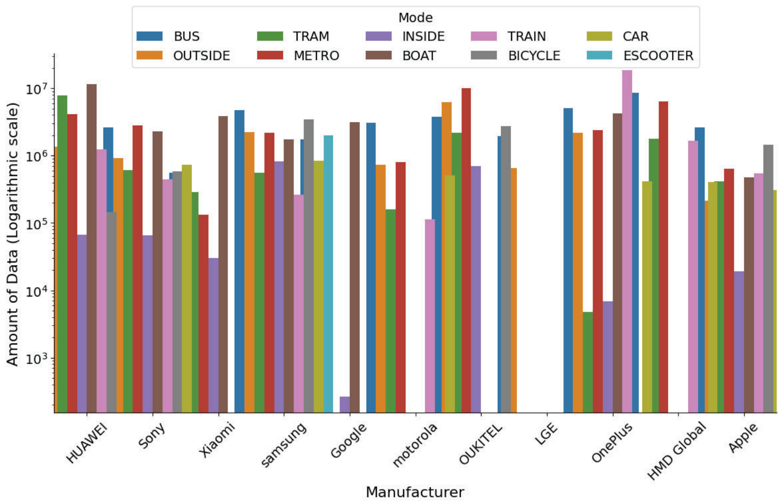


Figure 4.1: Distribution of data collected through each phone manufacturer.

NOR-TMD serves as a comprehensive resource for Smart Mobility research, supporting the training of machine learning models for multiclass classification of transportation modes. The structure and content of NOR-TMD were designed to enhance generalisability across devices and contexts, addressing issues of device homogeneity and limited sensor data that often restrict existing datasets. As the foundational resource for the experiments and evaluations conducted throughout this research, this dataset has demonstrated high utility in enabling robust model training for transport mode classification across diverse algorithms (Articles II-V).

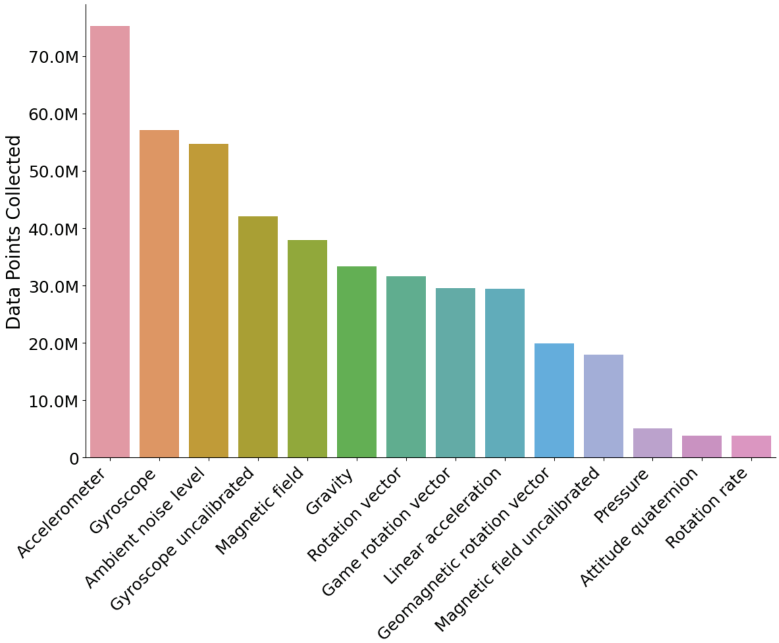


Figure 4.2: Distribution of data collected for each sensor.

In terms of temporal coverage, the dataset spans two separate 1-month periods, capturing different times of the day, from morning rush hours to late-night periods. This ensures that the dataset includes a wide variety of transportation scenarios, such as peak traffic hours, off-peak hours, and variations in transport availability. Geographically, the data was collected in two distinct cities in Norway, providing a mix of urban and suburban transport environments. This temporal and spatial diversity makes the dataset suitable for studying transport mode detection across a range of real-life settings, both in terms of daily commuting patterns and geographic variability. Figure 4.3 illustrates the total amount of data collected for each hour of the day, along with the distribution density of the collected data. The line plot shows the exact number of data points collected each hour, highlighting specific periods of high activity, such as between 5 and 9 in the morning, followed by a sharp drop between 9 and 15, which corresponds to regular work hours. This plot reflects the raw data and provides a detailed view of data collection trends. In contrast, the Kernel Density Estimate plot (right side) offers a generalised view of the overall trends in the data collection density across time.

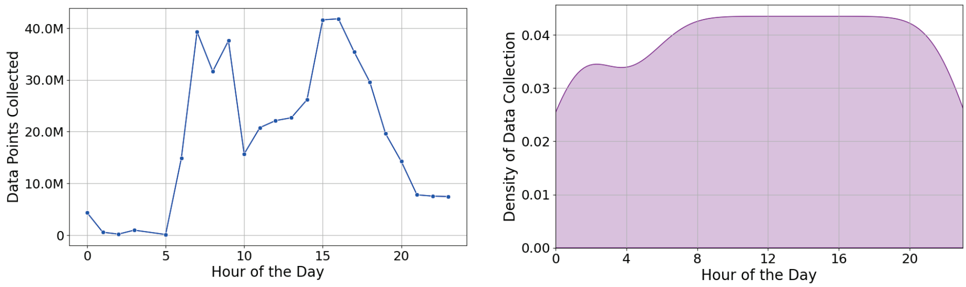


Figure 4.3: Data collected each hour of the day.

Although this dataset was primarily designed for transport mode detection, it has potential applications across various research areas. The sensor data could be utilised to identify patterns in movement and surroundings, making it adaptable for applications such as general activity recognition. Additionally, urban planners may find value in using the dataset to analyse public transportation usage patterns, providing insights to improve transport infrastructure. The temporal and geographic diversity of NOR-TMD also makes it suitable for applications in health monitoring, such as examining daily movement trends and identifying shifts in physical activity over time.

The NOR-TMD dataset developed in this design cycle is a significant contribution that forms the empirical foundation for this thesis. By addressing limitations in existing datasets and providing extensive, diverse, and realistic data, NOR-TMD directly supports the investigation of RQ1, RQ2, and RQ3, and thereby contributes to answering the overarching RQ0. The dataset underpins the experiments and evaluations presented in Articles II to V, enabling accurate, generalisable, and efficient transport mode detection. The relationships between articles, research questions, and key contributions are detailed in Tables 4.1, 4.2, and 4.3, which provide a comprehensive overview of how this dataset supports the research objectives across design cycles.

4.4 Design Cycle 3: Multilayer Perceptron

Building on the foundational dataset developed in Design Cycle 2, the third design cycle focuses on assessing the feasibility of using the collected sensor data for transport mode detection. This is addressed through two primary objectives: (1) identifying the most relevant features for transport mode detection using an ensemble of feature evaluation techniques, and (2) developing an efficient, high-accuracy multilayer perceptron (MLP) model tailored for on-device transport mode classification of Android sensor data. Consistent with the previous design cycle, this cycle adopted an objective-centred approach, as it marked the initial attempt at developing a model for sensor-based transport mode detection (Figure 3.2). This cycle sought to address the challenges of feature selection and model design by integrating diverse methods to evaluate feature importance and deploying lightweight yet effective neural network architectures. The results from this cycle provide valuable insights into feature relevance and demonstrate the feasibility of implementing local, on-device transport mode detection.

4.4.1 Feature Evaluation

To systematically identify the most relevant features for transport mode detection, this cycle employed an ensemble of feature importance techniques. The absence of a standardised method for evaluating feature importance in this domain necessitated an innovative approach leveraging multiple complementary algorithms. The ensemble included Classification and Regression Trees (CART) [220], Random Forest [221], XGBoost [222], ANOVA f-tests [223], and Mutual Information [224]. As each technique generates outputs on varying scales, individual thresholds were established to isolate the most relevant features for each approach. Thresholds were selected based on observed distribution patterns within each technique's results, allowing for a clear separation between features of high and low importance. Features exceeding these thresholds were combined to form a final feature set, which demonstrated the highest performance across models trained using this ensemble approach.

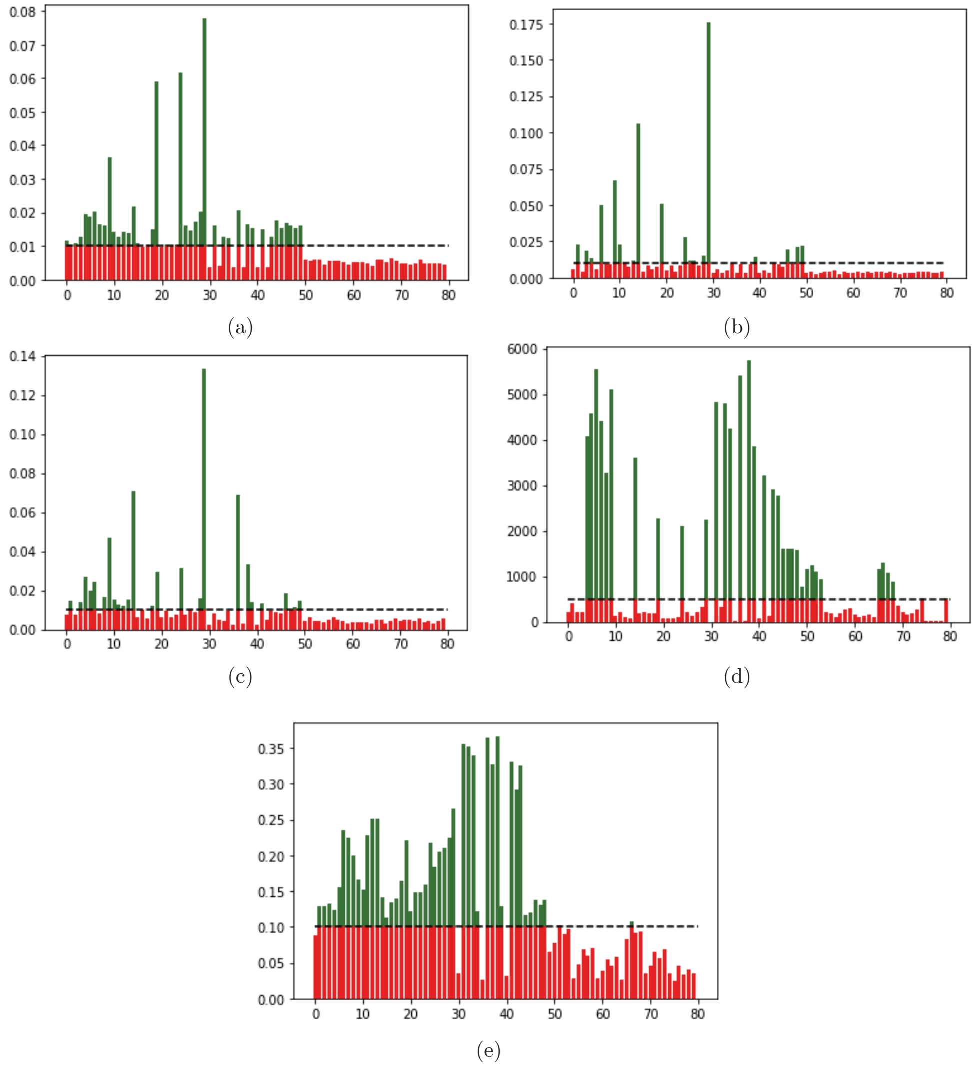


Figure 4.4: Feature importance analysis using different methods: (a) CART, (b) Random Forest, (c) XGBoost, (d) ANOVA f-test, and (e) Mutual Information. The X-axis represents features (indexed from 0 to 80), while the Y-axis shows the relative importance score assigned by each method.

Figure 4.4 illustrate the feature importance rankings generated by each technique, highlighting their unique contributions to the final feature set. The ensemble approach yielded higher accuracy than individual feature importance techniques; however, as thresholds were determined intuitively and the approach was only tested on Android data, further refinements in terms of normalisation and evaluation was necessary to more systematically identify the best features in subsequent iterations.

4.4.2 Model Development

The primary focus of this design cycle was the development and evaluation of a lightweight, multilayer perceptron (MLP) model capable of performing transport mode detection on Android devices. The choice of MLP was guided by the algorithm's balance between computational efficiency and predictive performance, making it well-suited for deployment in resource-constrained environments such as smartphones. The dataset used for model training and evaluation was based on NOR-TMD and included features derived from common smartphone sensors, such as the accelerometer, gyroscope, magnetometer, pressure sensor, rotation vector, and game rotation vector. Raw sensor readings were then aggregated using statistical functions, including mean, standard deviation, and quartile values, to generate features optimised for classification tasks. The model was trained to classify multiple transport modes, including bus, metro, train, tram, and a general other category encompassing walking, bicycling, and car travel. This classification scheme was designed to reflect the study's focus on public transport, grouping walking, bicycling, and car travel under a single other category to simplify the detection task and emphasise distinctions between public transport modes.

	Precision	Recall	F1-Score	Support
BUS	0.91	0.96	0.93	19331
METRO	0.90	0.89	0.90	10271
OTHER	0.93	0.92	0.92	9428
TRAIN	0.83	0.65	0.73	1174
TRAM	0.79	0.67	0.73	2752
Accuracy			0.91	42956
Micro Avg	0.87	0.82	0.84	42956
Weighted Avg	0.90	0.91	0.90	42956

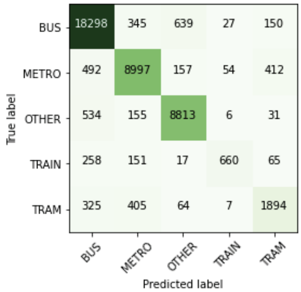


Figure 4.5: Classification report and confusion matrix for local model.

The resulting MLP model achieved an accuracy of 90.6% on unseen data collected in real-life settings, effectively distinguishing between transport modes across the defined categories. Figure 4.5 presents the classification report, which includes precision, recall, and F1-scores, as well as a confusion matrix highlighting classification performance across all transport modes. The model demonstrated particularly high precision and recall for classes with large amounts of test samples (support), such as bus, metro, and other. These results strengthen existing works that employ MLP architectures by confirming their utility for transport mode detection while also attesting to the applicability and robustness of the curated dataset proposed in this thesis, NOR-TMD, as a foundation for developing effective transport mode classification models. Furthermore, the lightweight design ensures compatibility with frameworks, such as TensorFlow Lite, facilitating seamless deployment on Android smartphones. This enables fully local processing, thereby supporting privacy-preserving solutions by eliminating the need to transmit sensitive data to external servers.

This design cycle makes critical contributions toward answering the core research questions by advancing feature evaluation methods (addressing RQ2) and demonstrating the feasibility of efficient, high-accuracy transport mode classification through the multilayer perceptron (addressing RQ1). The work conducted here is documented in Article II and lays the groundwork for subsequent cycles by providing both a systematic feature selection approach and a practical model architecture tailored for on-device deployment on Android devices. Together, these advances directly support the overall goal of efficient, generalisable, and privacy-preserving transport mode detection outlined in RQ0 and RQ3.

4.5 Design Cycle 4: Extreme Gradient Boosting

The fourth design cycle aimed to explore the use of Extreme Gradient Boosting (XGBoost), a tree-based ensemble learning algorithm, for transport mode detection. Building on the results and insights from the previous cycles, this iteration deliberately adopted a fundamentally different approach to feature evaluation to assess whether alternative methodologies could enhance model performance. Additionally, the scope of evaluation was broadened to include a more diverse range of transport modes. The primary goals of this cycle were twofold: (1) systematically explore feature importance using feature ablation and (2) develop high-accuracy transport mode detection models for both Android and iOS platforms. Building on the results of the previous cycle, this phase can be regarded as a refinement of the initial artefact developed earlier. Consequently, this design cycle commenced with a design and development centred initiation (Figure 3.2) to derive an enhanced artefact for transport mode detection.

4.5.1 Feature Evaluation

In contrast to the ensemble-based feature selection method used in earlier cycles, this cycle employed a feature ablation approach to evaluate the individual contributions of sensors and aggregation functions to model accuracy. This approach systematically removed individual features, one at a time, to measure their impact on model performance. The goal was to provide direct insights into the relevance of each feature and identify any that might adversely affect the model's effectiveness. The feature ablation process involved training an initial model using the full feature set as a baseline for comparison. Models were then trained with one sensor or aggregation function removed in turn, and the resulting accuracy differences were recorded. As shown in Table 4.5, removing individual sensors or aggregation functions generally resulted in only minor changes to accuracy, with the notable exception of the magnetometer on iOS, whose removal caused a significant accuracy drop. This indicates that while the model can maintain high accuracy without certain features, it often compensates for the removal of a feature because other sensors and aggregation functions capture related or overlapping information. The results underscore that the overall performance depends on the combined presence of multiple features rather than the criticality of any single one, with the magnetometer on iOS being an exception, indicating its unique role in differentiating transport modes on iOS devices.

Table 4.5: Model accuracy (%) after sensor and aggregation function removal.

Sensor Removed	**Android**	**iOS**	**Function Removed**	**Android**	**iOS**
Accelerometer	96.68	96.99	Min	96.60	97.02
Magnetometer	95.26	87.51	Max	96.80	97.01
Orientation	96.90	N/A	1st quantile	96.75	96.98
Gyroscope	95.66	97.05	2nd quantile	96.74	97.12
Barometer	95.99	95.87	3rd quantile	96.71	96.97
Gravity	96.82	97.08	Average	96.73	97.12
Linear acceleration	96.84	N/A	Range	96.91	97.18
Rotation vector	96.76	N/A	Variance	96.89	97.01
Game rot. vector	96.92	N/A	Standard deviation	96.76	97.19
Motion quaternion	N/A	96.73			
Rotation rate	N/A	97.11			
Audio	96.49	96.91			
All features	96.86	97.08			

In addition to feature ablation, aggregation functions were evaluated by training separate XGBoost models using only one aggregation function at a time. This approach was taken because the ablation study revealed only minimal differences in accuracy, making it challenging to draw clear conclusions about the contribution of individual aggregation functions. The results in Table 4.6 highlight that certain functions, such as minimum, maximum (Android), and 1st quantile (iOS), captured key variability in the data and delivered higher accuracies. Conversely, functions like variance and standard deviation were less informative, with the range function notably reducing accuracy on iOS. Despite these findings, retaining all aggregation functions in the final models proved beneficial, as their combined use consistently yielded the highest overall accuracy.

Table 4.6: Aggregation function experimentation.

Aggregation function	Android model accuracy %	iOS model accuracy %
Min	94.58	94.76
Max	94.97	95.23
1st quantile	92.71	95.50
2nd quantile	92.98	95.25
3rd quantile	93.26	95.27
Average	92.34	94.78
Range	92.06	84.73
Variance	87.03	82.89
Standard deviation	86.95	82.83

The feature ablation results provided valuable insights into the utility of individual sensors and aggregation functions, highlighting platform-specific differences and reinforcing the need for a comprehensive, adaptable approach to feature evaluation. This experiment underscored the exploratory nature of this design cycle, paving the way for refinements in subsequent iterations.

4.5.2 Model Development

During the work on Article III, XGBoost was selected due to its promising performance reported in the literature. Two models were developed using the XGBoost algorithm: one trained on data from Android devices and another on data from iOS devices. The goal of this experiment was to determine the highest achievable accuracy based on the collected data. To this end, audio was incorporated, as well as synthesising the minority classes using SMOTE. The data was preprocessed according to the procedures detailed in Chapter 3 in order to generate features for the model. In this iteration, the whole spectrum of transportation modes was utilised, including the following modes: "bus", "metro", "train", "tram", "other", "bicycle", "boat", and "car". In addition, the model trained on Android data also incorporated "e-scooter". The mode "e-scooter" was not present in the iOS data. Table 4.7 show the classification report of the two models, including precision, recall, and F1-score. Figures 4.6 and 4.7 display the confusion matrices for the two final models trained on data from Android and iOS devices, respectively. Both models were evaluated on unseen data, consisting of 30% of the initial data collection.

Table 4.7: Classification reports for XGBoost.

	Android				iOS			
	Precision	Recall	F1-score	Support	Precision	Recall	F1-score	Support
BICYCLE	0.99	1.00	1.00	16523	1.00	1.00	1.00	23183
BOAT	1.00	1.00	1.00	16669	1.00	1.00	1.00	23025
BUS	0.98	0.99	0.98	16497	0.98	0.97	0.97	23112
CAR	1.00	1.00	1.00	16465	0.99	0.99	0.99	23304
E-SCOOTER	1.00	1.00	1.00	16428	-	-	-	-
METRO	0.98	0.97	0.97	16477	0.96	0.96	0.96	22799
OTHER	0.99	0.99	0.99	16596	0.99	1.00	0.99	22898
TRAIN	0.99	0.99	0.99	16501	0.97	0.97	0.97	23077
TRAM	0.97	0.98	0.98	16603	0.95	0.96	0.96	23160
Accuracy			0.99	148759			0.98	184558
Micro avg	0.99	0.99	0.99	148759	0.98	0.98	0.98	184558
Weighted avg	0.99	0.99	0.99	148759	0.98	0.98	0.98	184558

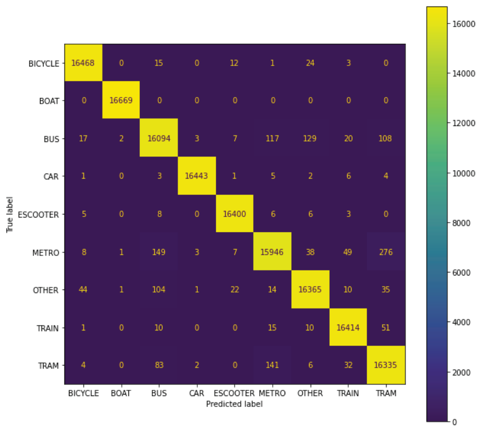


Figure 4.6: Confusion matrix Android.

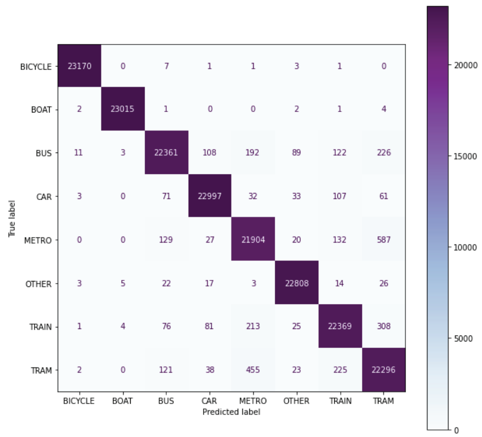


Figure 4.7: Confusion matrix iOS.

The Android model achieved an accuracy of 98.91%, while the final iOS model reached 98.03%. Although the primary focus of this research is on local transport mode detection, the XGBoost models are likely more suitable for deployment in centralised environments rather than directly on mobile devices. While no theoretical barriers prevent XGBoost from running on mobile hardware, current mobile machine learning frameworks, such as TensorFlow Lite, does not support it natively. TensorFlow Lite is optimised for running deep learning models on mobile and embedded devices, rather than tree-based models like XGBoost. Given that XGBoost models can require considerable memory and computational resources depending on the number and depth of trees they may not be optimal for direct deployment on resource-constrained mobile devices. Instead, XGBoost may be better suited for deployment on more computationally powerful servers, instead providing transport mode detection as a network-based service.

This design cycle makes several important contributions to the thesis. First, it advances the development of high-accuracy transport mode detection models by applying the Extreme Gradient Boosting (XGBoost) algorithm, directly addressing RQ1. Second, it contributes novel insights into feature relevance and sensor importance through systematic feature ablation and aggregation function experiments, supporting RQ2. Third, it demonstrates the adaptability and generalisability of the approach by developing separate models for both Android and iOS platforms, which relates to RQ3. The work presented in this cycle forms the basis of Article III and complements earlier design cycles by refining the artefact to improve accuracy and extend transport mode coverage.

4.6 Design Cycle 5: Ensemble Feature Ranking Framework

A significant contribution of this thesis is the development of a generic framework for feature evaluation and reduction (EFR-TMD), designed to identify the most relevant features for transport mode detection using mobile sensor data (Article IV). Throughout this research period, feature importance has been consistently assessed and analysed. Insights from previous design cycles highlighted the need for a more structured approach to feature evaluation and reduction.

Given that this framework represents a novel artefact aimed at addressing the overarching problem, this cycle was initiated with an objective-centred approach (Figure 3.2). In the second design cycle, an initial ensemble approach was tested, and the approach yielded positive results. In the third design cycle, an ablation approach was employed. While the model trained using this method achieved high accuracy, the feature importance experiments yielded inconclusive results. To address the limited conclusiveness of the ablation study, the ensemble approach explored in the second design cycle was revisited and refined. The goal of the fifth design cycle was therefore to enhance the ensemble method into a robust, generic framework capable of determining feature importance for transport mode detection across various algorithms. This refined framework built on the insights gained from previous cycles, aims to improve the adaptability and accuracy of feature selection across different machine learning algorithms. Figure 4.8 provides a high-level overview of the framework, illustrating the data flow and integration of the various components.

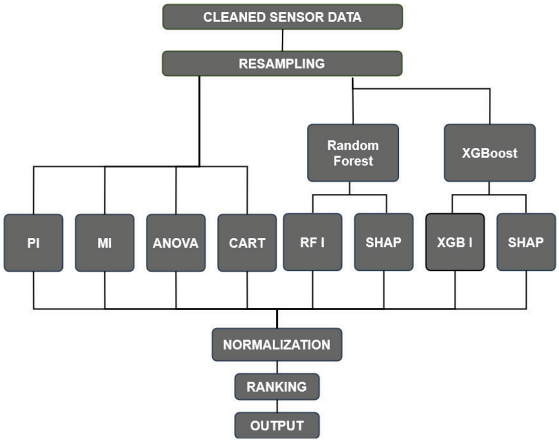


Figure 4.8: High-Level overview of EFR-TMD.

The framework integrates an ensemble of several feature importance techniques, normalises their output, and calculates the intersection of importance across all outputs to produce an ordered list of feature importance. The feature importance techniques included in the ensemble are Permutation Importance (PI), SHapley Additive exPlanations (SHAP), Mutual Information (MI), Analysis of Variance (ANOVA) f-test, and Classification and Regression Trees (CART), in addition to the built-in feature importance capabilities of the XGBoost and Random Forest algorithms. Since each feature importance technique produces results on different scales and metrics, these relative importance scores were normalised by assigning the feature with the highest importance score in each technique a value of one, while the feature with the lowest score was assigned a value equivalent to the total number of features. The normalised results were then intersected across all feature importance techniques to generate ranked lists of the most important sensors, aggregation functions, and their combinations.

Despite its simplicity, this framework effectively analyses the importance of both sensors and aggregation functions using data from Android and iOS devices. All sensors included in the NOR-TMD dataset were utilised, with their data aggregated using each of the statistical functions listed in Table 4.8. Figures 4.9 and 4.10 show the results in terms of sensors when applying the NOR-TMD dataset to the EFR-TMD framework for data derived from Android and iOS devices, respectively. A lower score indicates higher importance for the given sensor.

Table 4.8: Statistical aggregations applied to each sensor.

Aggregation Function	**Description**
Minimum	Smallest value in the segment
Maximum	Largest value in the segment
Average	Mean value in the segment
Range	Difference between maximum and minimum
Variance	Variability in the segment
Standard Deviation	Dispersion around the mean
Kurtosis	Measure of data distribution shape
1st Quartile	25th percentile of the data
2nd Quartile (Median)	50th percentile of the data
3rd Quartile	75th percentile of the data
Interquartile Range	Range between 1st and 3rd quartiles

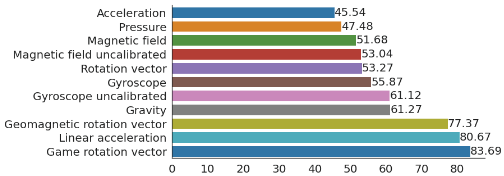


Figure 4.9: Rank of each Android sensor across all feature importance methods based on NOR-TMD.

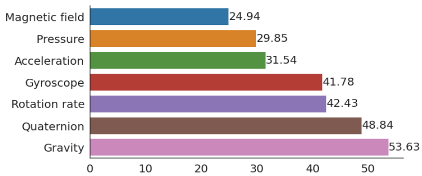


Figure 4.10: Rank of each iOS sensor across all feature importance methods based on NOR-TMD.

The Android sensors with the highest average rankings across all importance techniques include the accelerometer, pressure sensor, and magnetic field sensor. These three sensors are also identified as the most important on iOS, although their relative rankings differ slightly. The sensor rankings exhibit a significant range, indicating consistency among the techniques within the EFR-TMD framework in identifying the highest-ranked sensors. This suggests that the identified top sensors are likely to be the most applicable for transport mode detection. In contrast, a narrower range with closer rankings would imply greater variability across the techniques, potentially indicating a more arbitrary ranking. Although the range of importance scores is narrower for iOS sensors due to the smaller number of sensors available on iOS, the results still exhibit significant variability, further confirming the applicability of the top-ranked sensors.

Additionally, the EFR-TMD framework was employed to identify the most applicable aggregation functions. Figures 4.11 and 4.12 display the ranking of various popular statistical aggregation functions from the literature based on Android and iOS data, respectively. The importance of the aggregation functions varies more between Android and iOS. For Android data, the most significant aggregation functions are range, interquartile range, and standard deviation, whereas for iOS, the 1st quartile, minimum, and interquartile range are identified as the most important aggregation functions. The importance of the various aggregation functions on Android data shows relatively less variability, with the notable exception of kurtosis, which is identified as the least applicable aggregation function for transport mode detection. On iOS, however, there is greater variability, particularly in terms of the sensors identified as the two most important functions: the 1st quartile and minimum.

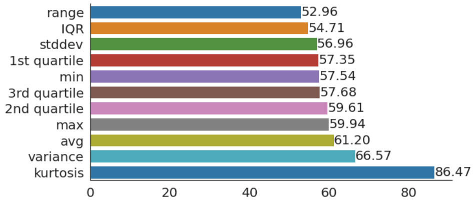


Figure 4.11: Average rank of each sensor across all feature importance methods using the iOS dataset.

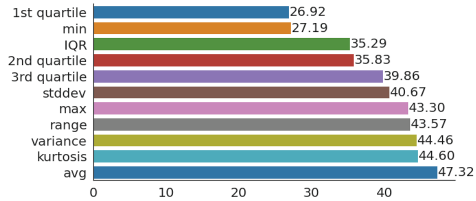


Figure 4.12: Average rank of each function across all feature importance methods using the iOS dataset.

To assess the generalisability of the findings from the feature importance evaluation across multiple algorithms, five commonly used algorithms for transport mode detection were employed: Convolutional Neural Network (CNN), Long-Short-Term Memory (LSTM), Multilayer Perceptron (MLP), XGBoost (XGB), and Random Forest (RF). The behaviour of the models in terms of accuracy, training time, inference time, and model size was observed as the bottom 25% of sensors and aggregation functions based on their importance scores were iteratively removed. This approach enabled an evaluation of the transferability of the results across different algorithms and provided insights into accuracy changes associated with the removal of specific sensors and functions. Table 4.9 presents the model performance results, showing the accuracy progression from models trained with all features to those trained using only the top 25% of features.

Table 4.9: Sensor and function ranking evaluation.

Algorithm	# Features	Inference Time (ms)	Training Time (m)	Model Size (MB)	Accuracy (%)
Android					
CNN (100%)	121	55.56	5.38	0.37	93.47
CNN (75%)	81	45.96	4.59	0.26	93.43
CNN (50%)	36	46.65	3.65	0.13	90.60
CNN (25%)	9	45.31	2.80	0.13	80.31
LSTM (100%)	121	47.32	69.36	0.08	93.77
LSTM (75%)	81	48.24	46.55	0.08	93.74
LSTM (50%)	36	47.58	21.25	0.08	93.84
LSTM (25%)	9	46.45	8.34	0.08	82.78
MLP (100%)	121	47.03	3.19	0.07	93.03
MLP (75%)	81	45.63	3.19	0.06	93.94
MLP (50%)	36	46.02	2.92	0.04	90.92
MLP (25%)	9	46.24	2.54	0.03	80.63
RF (100%)	121	9.34	1.15	45.53	96.55
RF (75%)	81	8.86	0.83	44.93	96.78
RF (50%)	36	8.23	0.57	58.01	95.13
RF (25%)	9	7.63	0.27	119.23	83.20
XGB (100%)	121	1.21	0.48	1.8	98.28
XGB (75%)	81	1.20	0.32	1.8	98.28
XGB (50%)	36	0.98	0.14	1.96	97.42
XGB (25%)	9	0.66	0.05	2.37	84.10
iOS					
CNN (100%)	77	48.64	9.35	0.25	77.49
CNN (75%)	54	52.89	8.78	0.18	79.11
CNN (50%)	24	49.47	7.73	0.09	76.73
CNN (25%)	6	46.7	6.32	0.09	67.09
LSTM (100%)	77	53.86	113.13	0.08	81.38
LSTM (75%)	54	54.26	163.2	0.08	87.77
LSTM (50%)	24	50.07	41.1	0.08	81.35
LSTM (25%)	6	46.67	16.02	0.08	72.28
MLP (100%)	77	51.22	7.37	0.06	79.51
MLP (75%)	54	51.69	7	0.05	77.14
MLP (50%)	24	48.49	6.46	0.04	76.59
MLP (25%)	6	44.94	6.37	0.03	67.07
RF (100%)	77	9.29	1.85	199.36	90.54
RF (75%)	54	8.57	1.63	212.99	89.08
RF (50%)	24	8.16	0.88	231.59	88.72
RF (25%)	6	7.69	0.42	348.03	81.76
XGB (100%)	77	1.3	0.62	2.72	93.06
XGB (75%)	54	1.4	0.53	2.71	91.11
XGB (50%)	24	1.12	0.24	2.63	89.47
XGB (25%)	6	0.81	0.1	2.56	78.5

Across the five algorithms, removing the bottom 25% of features resulted in negligible or no change in model accuracy. This indicates that the least important sensors and aggregation functions have limited relevance for transport mode detection. For the Android dataset, these features include the game rotation vector and linear acceleration sensors, along with the variance and kurtosis functions. On iOS, the least informative features are the gravity sensor and the average and kurtosis functions. Extending the removal to the bottom 50% of features also resulted in minimal accuracy loss across all models. On Android, this required excluding additional features such as the geomagnetic rotation vector, gravity sensor, uncalibrated gyroscope, and the second quartile, maximum, and mean functions. On iOS, this extended removal encompassed the following sensors: attitude quaternion and rotation rate, in addition to the maximum, range, and variance functions. Significant accuracy reduction occurred when retaining only the top 25% of features, likely due to the removal of critical sensors like the gyroscope on Android, and both the pressure and gyroscope sensors on iOS. The consistent accuracy reduction across the five algorithms suggests that the results of this feature ranking approach are both generalisable and transferable across different classes of algorithms.

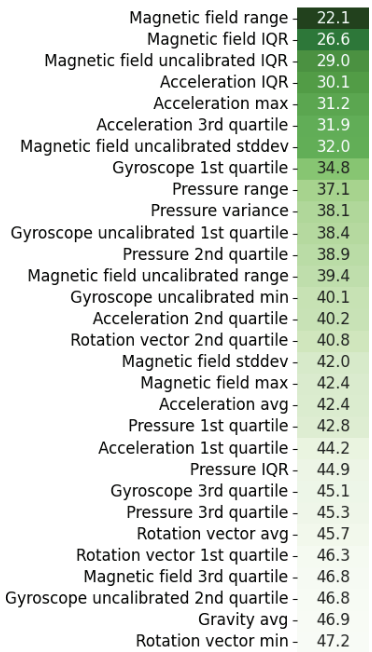


Figure 4.13: Top 30 most important composite sensors Android.

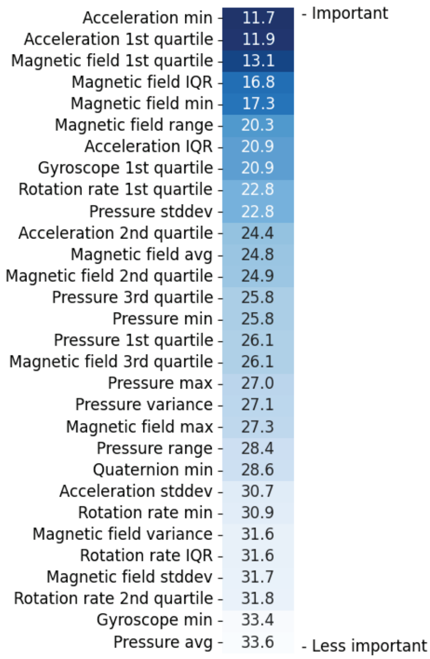


Figure 4.14: Top 30 most important composite sensors iOS.

As demonstrated in Table 4.9, it is possible to remove the bottom half of sensors and aggregation functions identified by the EFR-TMD framework while maintaining comparable accuracy. This consistency across all models highlights the robustness of the EFR-TMD framework in identifying universally important features for transport mode detection.

Table 4.10: Feature ranking evaluation results.

Algorithm	# Features	Inference Time (ms)	Training Time (m)	Model Size (MB)	Accuracy (%)
Android					
CNN (100%)	121	55.56	5.38	0.37	93.47
CNN (75%)	91	49.25	4.37	0.29	94.5
CNN (50%)	61	52.17	4.38	0.2	92.63
CNN (25%)	31	49.64	3.22	0.2	90.17
LSTM (100%)	121	47.32	69.36	0.08	93.77
LSTM (75%)	91	59.03	58.75	0.08	90.17
LSTM (50%)	61	52.34	41.07	0.08	91.19
LSTM (25%)	31	50.15	20.6	0.08	90.25
MLP (100%)	121	47.03	3.19	0.07	93.03
MLP (75%)	91	55.26	3.19	0.06	94.15
MLP (50%)	61	50.66	2.93	0.05	92.5
MLP (25%)	31	47.7	2.66	0.04	90.27
RF (100%)	121	9.34	1.15	45.5	96.55
RF (75%)	91	8.87	0.86	45.66	96.57
RF (50%)	61	8.34	0.66	49.18	96.16
RF (25%)	31	7.9	0.45	56.65	95.41
XGB (100%)	121	1.21	0.48	1.8	98.28
XGB (75%)	91	1.56	0.42	1.79	98.33
XGB (50%)	61	2.35	0.57	1.84	98.1
XGB (25%)	31	0.71	0.1	2.02	97.16
iOS					
CNN (100%)	77	48.64	9.35	0.25	77.49
CNN (75%)	58	53.97	9.37	0.19	78.15
CNN (50%)	39	50.65	8.38	0.13	78.81
CNN (25%)	20	43.43	6.74	0.13	77.31
LSTM (100%)	77	53.86	113.13	0.08	81.38
LSTM (75%)	58	56.8	88.93	0.08	82.4
LSTM (50%)	39	49.49	62.6	0.08	83.24
LSTM (25%)	20	43.42	31.41	0.08	80.6
MLP (100%)	77	51.22	7.37	0.06	79.51
MLP (75%)	58	53.41	7.37	0.05	78.55
MLP (50%)	39	46.25	7.37	0.04	77.96
MLP (25%)	20	43.54	5.91	0.03	76.76
RF (100%)	77	9.29	1.85	199.36	90.54
RF (75%)	58	8.7	1.59	203.7	89.87
RF (50%)	39	8.14	1.33	213.84	88.8
RF (25%)	20	7.73	0.83	227.14	89.08
XGB (100%)	77	1.3	0.62	2.72	93.06
XGB (75%)	58	1.42	0.56	2.7	92.35
XGB (50%)	39	1.18	0.33	2.68	90.79
XGB (25%)	20	0.83	0.17	2.63	88.86

Additionally, reducing features significantly decreases both training and inference times. The impact of feature reduction on model size depends on the algorithm. LSTMs maintain the same size regardless of feature dimensionality because their model complexity is driven by hidden units and layers rather than input size. In contrast, tree-based models like RF and XGBoost may increase in size as they compensate for fewer features by creating deeper or more complex trees. Neural networks such as CNNs and MLPs, however, exhibit consistent model size reduction when features are removed. To better understand the role of sensors and aggregation functions, the importance of features consisting of sensor data aggregated using specific functions (e.g., average acceleration or interquartile range of magnetic field values) were ranked. Figures 4.13 and 4.14 showcase the 30 most important composite features identified by EFR-TMD, where lower scores indicate higher importance. Table 4.10 further demonstrates the negligible impact on accuracy when the least relevant composite features are iteratively removed. This consistent behaviour across models reinforces the robustness of the EFR-TMD framework, showing that the least important features consistently have minimal relevance.

By ranking features holistically, EFR-TMD enables the removal of up to 75% of the lowest-ranked features with only minor drops in accuracy. This effectively reduces model dimensionality while preserving performance. The reduced dimensionality not only accelerates training and inference but also results in smaller models for certain neural network architectures, enhancing their suitability for deployment on resource-constrained devices. These findings were further validated through the development of a platform-agnostic framework for transport mode detection (Article V). By relying exclusively on the intersection of the top-ranked sensors and aggregation functions identified by EFR-TMD, and being validated in real-life scenarios, the framework provides further evidence of the method's practical relevance and robustness in identifying critical features, simplifying models, and improving computational efficiency.

This design cycle represents a pivotal advancement in addressing RQ2 and RQ3 by establishing a robust, model-agnostic feature evaluation and reduction framework that directly enhances the accuracy and efficiency of transport mode detection models. The EFR-TMD framework, developed and refined during this cycle, is comprehensively presented in Article IV and serves as a cornerstone for achieving lightweight, real-time inference crucial for on-device deployment, as further explored in Article V. By systematically identifying and prioritising the most relevant features, this cycle significantly contributes to the thesis's core contributions. Namely, the introduction of the EFR-TMD framework and the optimisation of transport mode detection models for practical, platform-agnostic mobile applications. Collectively, the work in this cycle not only advances theoretical understanding but also delivers tangible improvements in model generalisability, computational efficiency, and deployment readiness, thus making a critical contribution towards answering the overarching RQ0.

4.7 Design Cycle 6: Platform-Agnostic Framework for Local Transport Mode Detection

The aggregated results from the preceding design cycles provided the foundation for the sixth and final design cycle of this research. Building on the results derived from the EFR-TMD framework, as well as previous experimentation with multilayer perceptrons, this research introduces a lightweight, platform-agnostic framework designed for local transport mode detection, aimed at providing a privacy-preserving solution adaptable to diverse mobile platforms. Since the goal of this cycle was to design, deploy, and evaluate a lightweight, platform-agnostic framework, it adopted a design- and development-centred approach (Figure 3.2). From the results of the importance evaluation conducted in the previous design cycle it became evident that the most applicable sensors for transport mode detection were consistently effective across both Android and iOS devices. This finding, coupled with the ability to reduce the feature set substantially while preserving accuracy, prompted an investigation into the feasibility of a streamlined, platform-agnostic model for on-device transport mode detection (Article V).

Based on the previous experimentation, a Multilayer Perceptron (MLP) model was developed to infer transportation modes on both platforms, implemented and tested in real-life conditions on public transportation vehicles. The model was configured to classify transport modes including bus, metro, train, tram, and alternative modes (ALTM), with ALTM serving to differentiate public transport from other modes such as bicycle, e-scooter, boat, and car, equivalent to the previously mentioned other mode. Figures 4.15 and 4.16 present confusion matrices, while Table 4.11 displays the classification report derived from both simulated and real-life evaluations across Android and iOS devices.

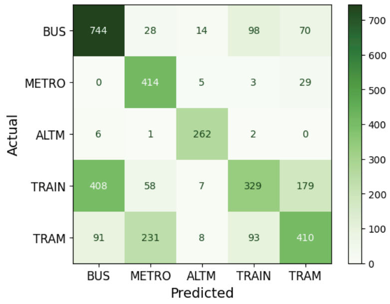


Figure 4.15: Confusion matrix of real-life experiment on Android devices.

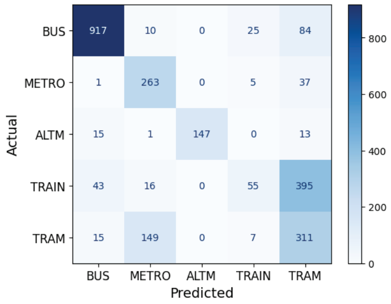


Figure 4.16: Confusion matrix of real-life experiment on iOS devices.

Class	**Precision**		**Recall**		**F1-Score**		**Support**	
	Android	iOS	Android	iOS	Android	iOS	Android	iOS
				Holdout set				
BUS	0.92	0.97	0.86	0.82	0.89	0.89	97846	116187
METRO	0.95	0.82	0.82	0.85	0.88	0.84	51346	24784
ALTM	0.93	0.86	0.86	0.86	0.89	0.86	59951	40960
TRAIN	0.35	0.53	0.89	0.86	0.51	0.66	5932	14742
TRAM	0.48	0.47	0.76	0.79	0.59	0.59	13300	11358
Accuracy					0.85	0.83	228375	208031
				Real-life Experiment				
BUS	0.60	0.93	0.78	0.89	0.68	0.90	954	1036
METRO	0.57	0.60	0.92	0.86	0.70	0.71	451	306
ALTM	0.89	1.00	0.97	0.84	0.92	0.91	271	176
TRAIN	0.63	0.60	0.34	0.11	0.44	0.18	981	509
TRAM	0.60	0.37	0.49	0.65	0.54	0.47	833	482
Accuracy					0.62	0.67	3490	2509

Table 4.11: Classification reports simulated vs. real-life experiment.

As illustrated, performance on the holdout set was significantly higher than in real-life tests, which suggests the possibility of overfitting, despite the extensive measures taken to mitigate it, as detailed in Chapter 3. Nevertheless, the objective of this cycle was not solely to maximise accuracy, but also to validate the feasibility of achieving comparable performance across multiple platforms with a single model configuration. The classification report shows that the model achieved similar results on both Android and iOS devices, suggesting a high degree of platform-agnostic compatibility. Beyond cross-platform consistency, a major objective of this cycle was to develop a compact, efficient model suitable for resource-constrained mobile devices.

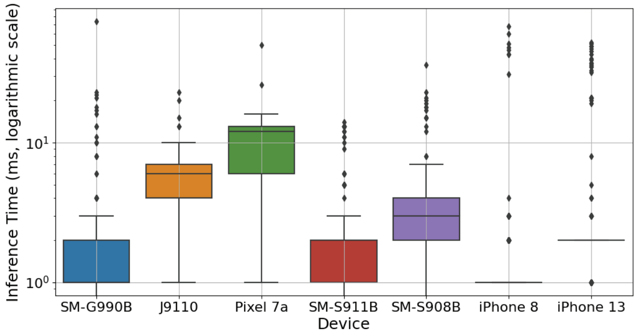


Figure 4.17: Inference time distribution across all devices.

Previous experiments demonstrated that dimensionality reduction could substantially decrease inference time, so the feature set was reduced to optimise speed and energy efficiency without compromising critical functionality. Figure 4.17 displays the inference time distribution across different devices, while Figure 4.18 compares inference times across transport modes. On Android devices, the average inference time was 5.31 milliseconds (ms), ranging from under 1 ms to 74 ms. For iOS devices, the average inference time was 2.05 ms, with a range from under 1 ms to 68 ms.

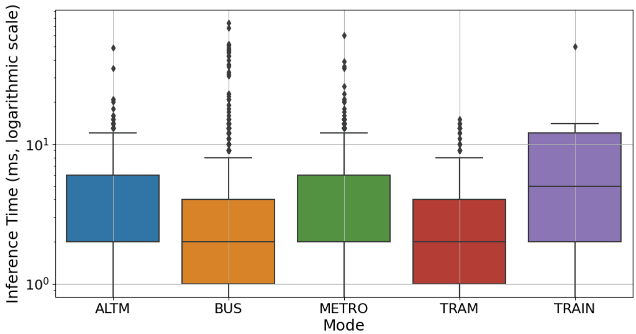


Figure 4.18: Inference time distribution across all modes.

Notably, while overall accuracy in a real-life context was lower than in controlled testing, misclassifications between public transportation modes and alternative modes (ALTM) were minimal. If all public transportation classes are grouped together, the model achieves a 99% accuracy rate in distinguishing between public and non-public transport modes. This high level of accuracy in identifying public transportation holds promise for applications such as automated ticketing, where accurately detecting whether a user is on board public transport vehicles is paramount.

Energy consumption is also of high importance when developing solutions tailored to mobile devices. While robust energy measurements were challenging, it was possible to gauge estimated energy consumption on Android devices using the Android Debug Bridge (ADB). Results indicated a low energy footprint for the model, with an estimated consumption range of 0.83% to 2.79% per hour during active testing. Given the similarity in model architecture and processing requirements across platforms, it is likely that this low energy impact observed on Android would similarly apply to iOS devices. The model's lightweight design, along with its minimal energy requirements, supports its suitability for extended, real-life use on mobile devices without imposing significant battery strain, making it a practical and efficient artefact for local transport mode detection.

This final design cycle demonstrates the practical deployment and real-world applicability of the research outcomes, directly addressing RQ0 and RQ3. Documented in Article V, the platform-agnostic framework validates the theoretical advances from previous cycles, particularly the effectiveness of the EFR-TMD feature reduction framework and optimised machine learning models, in operational mobile environments. The successful local deployment on both Android and iOS devices highlights the feasibility of real-time, low-latency transport mode detection with minimal energy overhead, fulfilling key contributions related to platform-agnostic, efficient, and generalisable on-device detection.

Chapter 5

Discussion

This chapter critically discusses the design artefacts developed in this research within the context of transport mode detection (TMD). Guided by the Design Science Research (DSR) paradigm, the study adopted an iterative process of designing, implementing, and evaluating solutions to address key challenges in TMD research. The primary gaps identified in the literature relate to the absence of benchmark datasets and standardised methods for feature selection and evaluation. Furthermore, while accuracy remains a central focus in TMD, it must increasingly be considered alongside computational efficiency and the feasibility of on-device inference. In addition, very few studies have investigated iOS data, possibly due to the lack of publicly available datasets containing such data. Finally, the cross-platform capabilities of on-device TMD remain largely unexplored. To address these challenges systematically, the following overarching research question was formulated to guide the design and evaluation of the artefacts developed in this study:

RQ0: How can efficient on-device, platform-agnostic transport mode detection be achieved on mobile devices?

To address distinct aspects of the overarching research aim, the following supporting questions were also explored:

RQ1: How can machine learning models achieve high accuracy in transport mode detection across diverse transport modes, ensuring generalisability in real-life applications?

RQ2: How can a standardised framework for feature evaluation and reduction systematically identify relevant features, ensuring consistency and enabling reliable feature reduction across machine learning models?

RQ3: How can transport mode detection models be optimised for real-time, low-latency inference while maintaining computational efficiency on mobile devices?

These research questions are addressed through the integrated outcomes of the various contributions that comprise this thesis. This research introduces high-accuracy machine learning models (Article II and Article III), presents a systematic framework for feature evaluation and reduction (Article IV), and culminates in the development of a deployable, platform-agnostic TMD framework (Article V). In addition, Article IV makes a significant dataset contribution to the research community through the release of NOR-TMD, thereby addressing a long-standing limitation in the field.

These contributions were developed and disseminated through five articles, each addressing distinct stages of the research process. Article I established the initial problem context by surveying existing Automated Fare Collection (AFC) systems, with the findings motivating a shift in focus towards TMD as an enabling component of AFC solutions. Article II explored the feasibility of developing models for TMD using the NOR-TMD dataset, while Article III concentrated on designing accurate and generalisable machine learning models for transport mode classification. Building upon the findings of Article II and Article III, Article IV introduced the EFR-TMD framework, an ensemble-based approach for feature evaluation and reduction. It also presented the comprehensive real-world dataset (NOR-TMD), which forms the foundation for all models developed in this thesis. Finally, Article V detailed the implementation of a lightweight, platform-agnostic framework for real-time, on-device inference, which was evaluated under real-life conditions using mobile devices on board various public transport vehicles. Throughout this chapter, each thematic discussion will reference the relevant article(s) underpinning the findings, clarifying and reiterating how the individual contributions integrate to answer the overarching research questions of this thesis.

This chapter provides a comprehensive discussion of the findings of this research. It begins by examining the development of high-accuracy transport mode detection models (Article III). The discussion then turns to the dataset introduced in this research (Article IV), highlighting its unique contributions and situating it in comparison with existing publicly available datasets. Subsequently, the proposed framework for feature evaluation and reduction (Article II and Article IV) is discussed, with particular emphasis on its role in standardising feature selection and enhancing computational efficiency. Finally, the chapter evaluates the development of the platform-agnostic framework, focusing on its optimisation strategies and performance in real-time, on-device inference settings (Article V). By situating these contributions within the broader context of existing research, this chapter demonstrates how the findings of this thesis advance the field of TMD, addressing critical challenges related to data availability, methodological consistency, and practical application.

5.1 High-Accuracy Transport Mode Detection

This research has iteratively sought to enhance computational efficiency while simultaneously incorporating a wider range of transportation modalities (Articles II-V), all while maintaining high classification accuracy. In alignment with the research questions posed in this thesis, an experiment was conducted to determine the highest level of accuracy achievable for transport mode detection, with a particular emphasis on integrating a broad spectrum of modes (Article III). Given the promising results reported in the literature for gradient-boosted trees [57], [151], [174], the XGBoost algorithm was selected for this purpose.

5.1.1 Extreme Gradient Boosting

By leveraging the full breadth of available features, this thesis achieved classification accuracies of 98.91% and 98.03% for Android and iOS data, respectively. To the best of current knowledge, the high-accuracy gradient-boosted models presented in this thesis (Article III) surpass existing approaches in transport mode detection. For instance, Nirmal, Disanayaka, Haputhanthri, *et al.* [57] achieved an accuracy of 94% despite incorporating only three transport modes: bus, train, and other. Additionally, their approach relied on GPS data, which is known to be highly energy-intensive [60], [77], [123] and raises privacy concerns [66], especially as their solution required remote processing on a centralised server.

Similarly, Aziz, Youkee, Ahmed, *et al.* [80] proposed an ensemble of different boosting algorithms, employing a majority voting mechanism to determine the final classification outcome. This approach expanded the approach of Nirmal, Disanayaka, Haputhanthri, *et al.* [57] by including five transportation modes while maintaining a similar accuracy (94%). The use of multiple models also significantly increases computational requirements.

Likewise, Lu, Pinaroc, Lv, *et al.* [151] attained an accuracy of approximately 94% when classifying between eight transport modes using only inertial sensor data. The models presented in this thesis, however, distinguish between eight (iOS) and nine (Android, including electric scooters) transport modes. Nevertheless, direct comparisons are challenging due to methodological differences; Lu, Pinaroc, Lv, *et al.* [151] employed frequency-domain features, whereas this thesis exclusively utilised time-domain features. Furthermore, while both studies classify a similar number of modes, the specific categories differ substantially. This thesis focuses on vehicular transportation, whereas Lu, Pinaroc, Lv, *et al.* [151] included human activities such as walking and running but omitted key transport modes such as trams and boats.

The accuracy achieved in this thesis, using XGBoost (Article III), reinforce existing evidence that XGBoost outperforms other commonly used algorithms, including random forests, decision trees, support vector machines, and multilayer perceptrons [57], [174]. The models developed in this thesis achieved high accuracies for both Android and iOS, placing them among the most accurate transport mode detection models to date, particularly considering the extensive range of transportation modes included.

Regardless of algorithmic or methodological approach, previous studies have typically achieved high accuracy but with a narrower selection of transport modes [14], [56], [75], [138], [143], [154]. This research distinguishes itself by accurately classifying a broader range of transport modes, including those rarely studied, such as trams and seagoing vessels. Most prior work in this domain, including approaches utilising neural networks and traditional classifiers, limits the number of classified modes, inherently reducing their practical applicability. By extending the range of recognised transport modes to include public transportation types such as trams and boats, this thesis addresses a key limitation in prior transport mode detection research, thereby offering a more holistic understanding of urban mobility. Furthermore, based on the reviewed literature, no other study has achieved comparable levels of accuracy using data collected from iOS devices.

5.1.2 Platform-Specific Model Development

Although this thesis strives to achieve platform-agnostic on-device transport mode detection, the findings related to the development of gradient boosted models (Article III) highlights the value of leveraging platform-specific data to optimise model performance. The decision to train separate models for Android and iOS was driven by the significant differences in hardware and sensor configurations across these platforms [48], [185]. The results demonstrate that transport mode detection can benefit from platform-specific modelling, as this approach captures nuances in sensor performance and data characteristics unique to each operating system. This differentiation between Android and iOS sensors, with their inherent variability, likely contributed to models that more effectively generalise across a diverse range of devices within each platform.

5.1.3 Practical Applications and Industry Integration

The high accuracy achieved by both models highlights their potential utility across a variety of practical applications, including automated fare collection (AFC), real-time public transportation analytics, and contextual user interfaces capable of adapting dynamically based on transport mode. Given that neural networks are predominantly the only models natively supported on mobile devices, models based on algorithms such as XGBoost are currently better suited for centralised implementations. The strong performance of the models for both Android and iOS data highlights their technological maturity and underscores their potential for integration into industry-level solutions aimed at enhancing the modelling of travel patterns and behaviours. Such integration could enable public transport operators, municipalities, and other stakeholders to make well-informed decisions regarding infrastructure expansion, route optimisation, and related initiatives.

5.1.4 Replacing Manual Surveys with Automated Detection

Understanding travel behaviours, travel demand, and the impact of transportation infrastructure on individuals is fundamental to transportation science. Traditionally, this type of data has been collected through travel surveys or diaries, where users report their travel motivations and modes of transportation. Methods such as face-to-face interviews, mail-out/mail-back paper diaries, phone interviews, and web forms have been commonly employed for this purpose [101]. However, automated transport mode detection (TMD) solutions offer a more efficient means of collecting this information, potentially reducing the need for resource-intensive, manual data collection methods. By implementing the high accuracy models presented in this thesis within a centralised architecture, there is no requirement for additional on-board equipment to infer the mode of transportation. Instead, a traveller's device could continuously stream sensor data, allowing public transport operators to determine the mode of travel in real-time. Additionally, the incorporation of GPS data, correlated with real-time public transport operator systems, could serve as a foundation for automated fare collection [14]. This research also demonstrates that XGBoost offers significantly lower inference times compared to alternative approaches, such as neural networks. Consequently, XGBoost may be particularly advantageous in centralised deployments, where its faster inference capabilities can free up computational resources more efficiently, thereby enabling a significantly greater number of classifications per time unit.

5.1.5 Limitations

While this contribution presents numerous benefits, it is also accompanied by challenges and limitations. The most significant limitation in the context of this research is the lack of native support for gradient boosted models on mobile devices, which inhibits the possibility of running them locally. Although centralised deployment offers distinct advantages from the perspective of public transport operators, there are notable downsides for travellers, particularly concerning privacy. A centralised architecture necessitates the continuous streaming of highly granular data about the user's movements and environment to public transport operators. If misused, this data could potentially reveal sensitive information about a traveller's health, habits, relationships, and other private details. For this reason, an important goal is to achieve on-device inference, thereby ensuring that data remains securely on the traveller's device.

Another limitation related to privacy is the inclusion of audio data. Audio data can be perceived as highly invasive, which is why mobile operating systems require explicit user permissions to enable software to access and record events from the microphone. This requirement complicates the application's usability and diminishes its pervasiveness, as travellers would need to actively interact with the application for it to function effectively. Furthermore, travellers may reasonably question why a public transportation solution would require access to the microphone, potentially leading to resistance in adopting the system.

Moreover, energy consumption was not investigated in relation to this approach (Article III). The choice of sensors has a significant influence on energy consumption [50]. In particular, activating the microphone has been found to consume considerably more energy compared to sensors like the accelerometer, which are known for their lower energy requirements [225], [226]. Although the models are not executed directly on the traveller's device, the use of the microphone, alongside continuous data transmission to a centralised system, could result in a considerable energy consumption. That being said, there are measures that can be implemented in order to mitigate this. For instance, data can be exchanged with the server in a more opportunistic way and the application can "piggyback" on other applications using already collected data to reduce the energy consumption [56].

A further limitation is that, while these models were evaluated using real-life data, they were not implemented or tested in an actual real-life setting. As demonstrated in this thesis and in previous studies [37], [56], the results obtained from real-life evaluations are often significantly worse compared to those derived under controlled simulated conditions. Therefore, the lack of real-life implementation presents a challenge to understanding the full practical potential of the proposed models. That being said, the results presented in this thesis help clarify which machine learning strategies are most effective in achieving high accuracy for transport mode detection across a broad range of transportation modes.

5.2 A Diverse and Representative Dataset for Mobility Research

This research introduces the NOR-TMD dataset (Article IV) as a design artefact, developed through the DSR process to provide a comprehensive and representative resource for mobility research. As a reusable instantiation, the dataset addresses the lack of diversity in existing resources by incorporating multiple devices, transportation modes, and operating systems, ensuring its applicability for both academic and industry use. The dataset marks a significant advancement in addressing the limitations of existing resources for transport mode detection (TMD).

Table 5.1: **Overview of publicly available sensor-based datasets.** N.M = Number of modes. N.S = Number of sensors. N.DL = Number of device placements during data collection. N.UD = Number of unique devices. N.UP = Number of unique participants.

Dataset	**N.M**	**N.S**	**OS**	**N.DL**	**N.UD**	**N.UP**	**Hours**
HTC	9	3	Android	N/A	1	224	8311h
SHL	8	16	Android	4	1	3	2812h
US-TMD	5	8	Android	N/A	11	13	31h
Collecty	8	4	Android	N/A	N/A	15	242h
NOR-TMD	10	13	Android, iOS	3	57	101	609h

Table 5.2: **Overview over publicly available sensor-based datasets**

Dataset	Modes	Sensors	Placement
HTC [123]	Still, Walk, Run, Bike, Motorcycle, Car, Bus, Metro, Train	Accelerometer, Magnetometer, Gyroscope	N/A
SHL [183]	Still, Walk, Run, Bike, Car, Bus, Metro, Train	Accelerometer, Magnetometer, Gyroscope, Orientation, Gravity, Linear acceleration, Ambient pressure, Google's activity recognition API, Ambient light, Battery level and temperature, Satellite reception, WiFi reception, Mobile phone cell reception, GPS, Audio	Hand, Hips, Torso, Backpack
US-TMD [72]	Still, Walk, Car, Bus, Train	Accelerometer, Magnetometer, Gyroscope, Gravity, Ambient light, Ambient pressure, Audio, Proximity	N/A
Collecty [184]	Walk, Run, Bike, Car, Bus, Train, Tram, E-scooter	Accelerometer, Magnetometer, Gyroscope, Linear acceleration	N/A
NOR-TMD	Bike, Metro, Train, Tram, Bus, Boat, Car, E-scooter, Inside, Outside	Accelerometer, Magnetometer, Uncalibrated magnetometer, Gyroscope, Uncalibrated gyroscope, Rotation vector, Game rotation vector, Rotation rate, Gravity, Quaternion, Linear acceleration, Ambient pressure, Built-in activity recognition	Hand, Pocket, Other

Notably, NOR-TMD excels in incorporating diversity across multiple dimensions, including devices, participants, geographical regions, device placements, platforms, sensors, and transportation modes, thereby providing a robust foundation for advancing TMD research. The NOR-TMD dataset represents a substantial advancement in addressing the limitations of existing resources for transport mode detection. A review of the literature identified only four publicly available datasets containing sensor data collected from various modes of transportation: the HTC dataset [123], the SHL dataset [183], the US-TMD dataset [72], and the Collecty dataset [184]. While each of these datasets offers valuable contributions, they also exhibit notable limitations. Tables 5.1 and 5.2 provides detailed comparisons of the key features of these four datasets alongside the proposed NOR-TMD dataset, highlighting its enhanced capabilities and broader scope.

5.2.1 Inclusion of iOS and Sensor Diversity

A significant contribution of the NOR-TMD dataset is the inclusion of sensor data collected from iOS devices. This feature distinguishes it from other datasets, which do not incorporate iOS data, possibly explaining why most related studies on transport mode detection have focused on Android platforms. This presents a substantial challenge, as iOS devices constitute a considerable share of the global mobile market [50] and developing practical and widely applicable solutions necessitates the inclusion of iOS data. The NOR-TMD dataset encompasses the most extensive range of smartphone sensors, incorporating all standard sensors available on both Android and iOS operating systems. While NOR-TMD includes a wider spectrum of motion and ambient sensor diversity by incorporating uncalibrated versions of sensors, as well as iOS sensors, the SHL dataset [183] includes additional data types, such as location data, ambient light, satellite reception, audio, battery level, and battery temperature.

Although data on ambient light, battery status, location, and audio was initially collected, these were excluded from the final dataset. Ambient light and battery information were omitted due to the lack of a clear relationship with transport mode detection. While prior work demonstrates the effectiveness of location data in transport mode detection [114]–[118], location data is not the primary focus of this research due to factors such as its high energy consumption [60], [77], [123] and privacy concerns [66]. Similarly, while audio data has demonstrated effectiveness in this context [141], it was not included in NOR-TMD due to privacy considerations.

In contrast to the NOR-TMD and SHL datasets, others such as HTC [123], US-TMD [72], and Collecty [184] lack key sensors. The HTC dataset [123], for instance, includes only the accelerometer, magnetometer, and gyroscope. While these sensors are widely used, in addition to being identified as highly relevant for transport mode detection by the EFR-TMD framework proposed in this thesis, the absence of barometer data is a notable limitation. The barometer has been shown to be highly useful in prior studies [135]–[138] and was identified as a critical sensor by EFR-TMD. The omission of barometer data in the HTC dataset can be attributed to its release in 2014, a time when most smartphones lacked this sensor. This underscores the need for updated datasets that reflect the sensor capabilities of modern devices. However, similar to the HTC dataset, the Collecty dataset [184] includes accelerometer, magnetometer and gyroscope, in addition to linear acceleration data, but it inexplicably omits the barometer despite the dataset being published as recently as 2023. In contrast, the US-TMD dataset [72] incorporates a broader set of sensors, including the barometer and audio.

While the SHL dataset [183] remains the most diverse in terms of data types, as it includes location and network data, the NOR-TMD dataset stands out as the most comprehensive dataset in terms of inertial sensor diversity. Its inclusion of iOS sensor data and broader sensor coverage underscores its contribution as a valuable resource for advancing transport mode detection research.

5.2.2 Transport Mode Coverage

Another critical aspect of these datasets, is the availability of transportation modes, as these directly influence the ability to classify and differentiate between different modes of transportation. Due to the lack of ground truth data many studies are focused on easy-to-detect modes [46] and the absence of benchmark datasets hinders comparisons of results [80]. Most of the datasets exhibit a decent variety of modes, with the exception of US-TMD [72], which only includes buses and trains as motorised modes. The Collecty dataset [184] expands this by including trams and electric scooters. Although the SHL dataset [183] and the HTC dataset [123] does not include trams, both datasets instead incorporate the modes car and metro. The HTC dataset [123] also includes motorcycles, making it the only dataset to feature this mode. The NOR-TMD dataset includes all the motorised modes present in the other datasets, with the exception of motorcycles, while adding seagoing vessels, which are an integral part of public transport in many countries [62].

As this research focuses on public transportation solutions, modes such as being still, walking, or running were excluded. Instead, a different, less granular approach was adopted, categorising modes as simply being inside or outside. These categories serve as useful differentiators for classifying various transportation modes. Considering the context of automated fare collection (AFC) solutions as an example, it is not as critical to distinguish whether a person is still, walking, or running. Instead, it is more important to discern whether a traveller is located within a specific mode of transportation and, therefore, subject to fare collection, or outside of a vehicle, where fare collection does not apply. While activities such as being still, walking, or running are important in the realm of activity recognition, they hold less relevance in the context of transport mode detection, particularly when focusing on public transportation. Among the datasets, NOR-TMD stands out as the most diverse in terms of public transportation modes. It incorporates all such modes from other datasets and uniquely includes data collected aboard ferries, making it the only dataset to offer this mode of transportation.

5.2.3 Participant, Device, and Placement Diversity

Another key factor is the diversity of participants and devices used to collect the dataset, as well as its size. The HTC dataset [123] has the largest number of participants, with 224 unique individuals collecting 8,311 hours of data. However, all participants utilised the same device for data collection, which poses significant challenges, as sensor data quality and variability differ across devices due to differences in hardware, operating systems, and manufacturer-specific sensor implementations [48], [185]. Similar to the HTC dataset [123], the SHL dataset [183] is also large in size but was collected by only three participants using the same device model. The US-TMD dataset [72] and the Collecty dataset [184] involve a larger number of unique participants, and while the US-TMD dataset [72] was collected using 11 unique devices, the authors of the Collecty dataset [184] do not specify the uniqueness of their devices, stating only that the 15 participants in their study used their personal devices for data collection. Device diversity is an important factor, as it enhances the dataset's generalisability [189], enabling robust transport mode detection models that can perform across a variety of hardware and software configurations.

Furthermore, participant diversity is also crucial, as different travellers exhibit distinct travel behaviours and patterns, which can vary significantly based on demographics [186] and individual habits [188]. In real-life scenarios, users interact with their devices in various ways, such as holding them in their hands, placing them in a pocket, or storing them in a bag. These different placements significantly influence sensor readings [61], [143]. Consequently, labelling the location of the device during data collection can provide valuable contextual information. Apart from the NOR-TMD dataset, the SHL dataset is the only other dataset that includes device location information during data collection. The SHL dataset provides four labels for device placement, whereas the NOR-TMD dataset offers three. During the collection of the SHL dataset Gjoreski, Ciliberto, Wang, *et al.* [183] adopted a highly controlled data collection protocol, ensuring that devices were only positioned in specific, predefined locations. While this strict methodology is reasonable, it could be argued that a more flexible approach, such as that used in the NOR-TMD dataset, may yield data that better reflects real-life usage as it is exceedingly challenging to systematically capture all possible device placements. The NOR-TMD dataset incorporates an "other" label, allowing participants to place devices in unconventional locations, such as backpacks, purses, or even the centre console of a car while driving. This approach potentially results in more diverse data, which may improve classification accuracy in real-life applications where users are likely to store devices in a variety of locations. In contrast, solutions developed using the SHL dataset may struggle to accurately classify modes of transportation when devices are stored in locations outside those specifically included in their controlled study.

5.2.4 Geographical Diversity

Lastly, geographical diversity is an important factor, as distinct travel patterns and behaviours can vary across regions [187]. While none of the discussed datasets include a large variety of geographical locations for data collection, some are less diverse in this regard than others. The HTC dataset exhibits limited geographical diversity, as it was collected from only two distinct roadways. In contrast, the SHL dataset encompasses a broader geographical range, primarily spanning from Brighton, UK, to London, UK. The study by Erdelić, Erdelić, and Carić [184] does not specify the devices used, or the geographical locations from which the Collecty dataset was obtained. Similarly, Carpineti, Lomonaco, Bedogni, *et al.* [72] provides no details regarding the geographical locations of the data collection which resulted in the US-TMD dataset. The NOR-TMD dataset includes data collected from two cities in Norway. While this is similar to the SHL dataset, the two geographical areas differ significantly. The SHL dataset covers regions in close proximity to one another, whereas the two cities in the NOR-TMD dataset, Oslo and Bodø, Norway, are much farther apart. Consequently, the NOR-TMD dataset can be considered more geographically diverse than the SHL dataset.

5.2.5 Limitations

Despite its strengths, the NOR-TMD dataset presents certain challenges that should be addressed in future research. One notable issue is the imbalance across users, devices, and transport modes. This imbalance poses challenges related to bias and the risk of overfitting, which must be carefully addressed depending on the specific use case of the dataset. During data collection, the primary objective was to obtain a comprehensive volume of data while ensuring that participants adhered to their regular travel behaviours, to curate a representative dataset. An alternative approach might have involved selecting participants specifically based on their travel behaviour and devices, to ensure a more balanced representation across transport modes and devices. However, such an approach would have been difficult to achieve practically, as all interested participants were invited to participate in the study. While the outside and inside labels provide useful contextual information, a more granular approach could enhance classification granularity. Initially, these labels were intended to separate outdoor transport modes, such as cycling or using an electrical scooter, from simply just being outside, as well as differentiate between being inside a building versus being inside a transport vehicle. While the outside and inside categories provided useful contextual information, a more granular set of labels might have enhanced the dataset's utility further. For example, differentiating activities such as *walking inside* versus *walking outside* or *sitting inside* versus *sitting outside* could offer a richer, more nuanced understanding of participant behaviour.

Moreover, using more detailed labels would facilitate better comparison with existing datasets and potentially allow for a more seamless integration with other available datasets, thus enhancing the dataset's overall applicability and generalisability. A further limitation is related to the issue of irregular sampling frequencies, which can have large impacts on the accuracy and efficiency of solutions for TMD [166]. The sensor data was collected using the native frameworks of the operating systems, with sensors being triggered based on changes, which resulted in non-uniform sampling intervals. This approach provided a detailed and dynamic representation of sensor variations, capturing fine-grained changes as they occurred. However, this also led to variability in the number of sensor events within each segment of data. Consequently, during both training and real-life implementation, the number of sensor data points per segment may differ from those used during the original model training phase. Such irregularity could potentially impact model performance, as models may not receive consistent volumes of input data across different scenarios, thus affecting reliability and robustness. This challenge points to the need for careful consideration of data preprocessing techniques, such as interpolation or segment standardisation, to mitigate the effects of variable sampling rates on model outcomes. Nevertheless, this dataset represents a significant contribution towards addressing all the research questions posed in this thesis, as it provides a foundational basis for developing comprehensive answers to these inquiries.

5.3 Feature Importance and Reduction for Transport Mode Detection

A key focus of this thesis was the development of a standardised framework for feature evaluation and reduction for transport mode detection research (Article IV). To this end, this thesis introduces the EFR-TMD framework (Ensemble Feature Ranking for Transport Mode Detection) as a prescriptive design artefact, developed through iterative Design Science Research (DSR) cycles (Article II and Article IV). EFR-TMD combines multiple feature selection techniques in an ensemble approach to produce consistent and interpretable rankings of feature importance across a range of machine learning models. By improving the robustness and transferability across an array of different algorithms, the framework supports more efficient TMD model development, contributing to ongoing efforts to establish and validate methodological standards in the field [45]. While designed specifically for TMD and evaluated within that context, the general principles underlying EFR-TMD may also be adaptable to other domains with similar feature selection challenges.

While the literature is converging on many preprocessing steps for preparing and analysing smartphone data for transport mode detection solutions, building to a large degree on practises from human activity recognition [59], the literature does not seem to agree on which features are the most applicable in this context. Features used in transport mode detection solutions mainly consists of sensor data aggregated or transformed using established mathematical functions from statistics or signal processing and given the wide array of sensors modern smartphones are equipped with, this leads to a wide array of options when selecting and extracting features. Investigating the literature holistically, nearly all combinations have been explored, although not systematically and very fragmented. This makes it difficult to pinpoint which features are most suited for new researchers in the field or practitioners seeking to build solutions based on the current state of the art in transport mode detection, leading to a large variation of utilised features. As a result, it becomes difficult to evaluate and compare different algorithmic approaches presented in the literature [80], as the features used influence the performance of the proposed solutions and overshadows the impact of other novel contributions relating to architectures, algorithms, or processing techniques.

5.3.1 EFR-TMD: A Robust Ensemble-Based Approach

Existing approaches to feature extraction and selection often rely on experimentation, domain expertise, or intuition [57], [85], [135]. Some studies employ ablation analysis, where a model is trained on the full feature set, and features are systematically removed until a significant drop in accuracy is observed [60]. Other studies employ sequential forward selection (SFS) which operates in the opposite manner, starting with an empty feature set and iteratively adding features until further additions fail to improve accuracy [54]. However, methods that iteratively add or remove features are limited in their generalisability, as they typically indicate the relevance of features for a specific algorithm but do not necessarily translate well to other contexts or algorithms. Estimating the importance of features in machine learning is currently very unreliable [178] as the different methods can yield very different results, even when applied on the same dataset [175]. Previous work argues that selecting features based on intuition or domain knowledge can sometimes achieve results comparable to those obtained through formal feature selection methods [82]. However, this reliance on intuition and domain knowledge attests more to the lack of reliable feature importance approaches than suggesting that selecting features based on domain knowledge is a better approach.

To address this challenge this research presents an ensemble approach that combine various methods, normalise the output and provide a more general importance score across the incorporated techniques. Although simplistic, the framework has been thoroughly evaluated across the most popular algorithmic approaches for transport mode detection (Tables 4.9 and 4.10). The results of the evaluation clearly demonstrates that removing the features deemed less applicable by EFR-TMD does not lead to a loss of accuracy. On the contrary, in some cases a slight accuracy increase can be observed, further highlighting the capabilities of the framework to identify features that are not applicable in the context of transport mode detection. Furthermore, the evaluation results exhibit remarkable consistency across very different algorithmic approaches which attests to the generalisability of the framework and indicates that the results of the feature importance can be seen as generic.

5.3.2 Insights from Sensor and Aggregation Function Evaluation

Based on the feature importance results, average importance scores were computed for individual sensors and aggregation functions. The EFR-TMD framework aggregates the outputs of several feature importance methods, and the distribution of scores across features provides insight into the level of agreement among these methods. A wide range of scores, where some features receive consistently high rankings and others receive consistently low rankings, indicates that the methods agree on which features are more or less important. This agreement results in a clearer separation between important and unimportant features. In contrast, when the scores for all features are relatively similar, it suggests that the methods disagree, with some ranking a feature highly and others ranking it poorly. These conflicting evaluations tend to average out, which reduces the ability of the framework to reliably distinguish between features. Therefore, a broader range of importance scores implies stronger consensus and increased confidence in the resulting rankings.

Using the EFR-TMD approach, the most applicable sensors for transport mode detection across both operating systems were identified as the accelerometer, the magnetometer, and the barometer (Figures 4.9 and 4.10). The range of importance scores across sensors is sufficiently wide, indicating a clear distinction between the most and least relevant sensors. This suggests that the top-ranked sensors, including the accelerometer, magnetometer, and barometer, are consistently identified as important across multiple techniques within the ensemble framework. This finding is consistent with prior research, where the inclusion of these sensors has been shown to yield strong performance in transport mode detection [60], [85], [135]. These sensors have not only demonstrated practical applicability, but there are also clear reasons grounded in domain knowledge that support their use. For instance, acceleration and deceleration patterns tend to be consistent within the same transportation mode [60]. The magnetometer captures changes in the magnetic field that are influenced by the vehicle's structure [85]. In addition, the pressure sensor is sensitive enough to detect subtle road irregularities, which can help distinguish between different modes of transportation [135]. To the best of current understanding, there are no attempts at systematically evaluating the applicability of smartphone sensors in the context of transport mode detection and the feature evaluation results thus reinforces the current indication as to which sensors are more applicable in this context. Furthermore, the fact that the proposed framework (Article IV) identified sensors that already have proven utility in previous research further attests to the utility of this framework. These results also makes the possibilities of developing cross-platform solutions for

Android and iOS explicit as the same top-ranked sensors were identified as the most applicable across both operating systems.

For aggregation functions (Figures 4.11 and 4.12), the overall range of importance scores is relatively narrow. This limited spread reduces the reliability of the function ranking results. However, there are some notable exceptions. For example, the kurtosis function on Android data is consistently ranked significantly lower than other functions, indicating limited applicability for transport mode detection. Furthermore, the difference in scores between the range function and both the variance and average functions is considerable. This suggests that, in general, variance and average are less informative in this context. In contrast, the results derived from iOS data display a broader spread in importance scores, which indicates a clearer distinction among the functions and suggests that the most highly ranked functions are more effective when applied to features derived from iOS.

Looking at the results of the ranking of composite features (Figures 4.13 and 4.14), it is evident that different functions are more applicable given specific sensors. For instance, while the range function is deemed the most applicable aggregation function for magnetic field data derived from Android devices, the interquartile range function is deemed the most important function for acceleration data. Similarly, based on data from iOS devices, the minimum function is considered the most important function for acceleration, while the 1st quartile is deemed the most important function for magnetic field data. It is difficult to target the exact cause of this discrepancy, however one potential explanation can be attested to differences in hardware and sensor implementations between the two operating systems [48], [185]. For instance, accelerometers on Android measure in meters per second squared (m/s^2), while iOS devices use gravitational increments. These discrepancies can influence aggregation function effectiveness, as seen in the higher importance of quartile-based functions for iOS data.

Another possible explanation is differences in sampling frequencies, as a lower frequency leads to less values being aggregated within each segment, which in turn leads to lower accuracies of classifiers [166]. During data collection a sampling frequency of 5 Hz (every 200 ms) was defined. However due to variations in operating system resource management, hardware differences, and the operating system's sensor prioritisation, the actual sampling frequency varied. Many previous studies prioritise maximising sampling frequency over ensuring a consistent sampling rate [94], [147]. Similarly, in this research, a high sampling frequency was prioritised to maximise data collection, rather than maintaining consistency across operating systems. Furthermore, in order to address class imbalances the data was re-sampled using SMOTE [170], which might further have contributed to this discrepancy.

5.3.3 Implications for Model Performance and Sustainability

The development of a framework for feature importance and dimensionality reduction also have direct implications for practitioners seeking to deploy transport mode detection systems. By systematically identifying the most critical features, EFR-TMD provides a structured approach to feature reduction, potentially improving classification performance [153] and enabling models to operate more efficiently. The results presented in this research indicate that up to 75% of commonly used features in the literature can be eliminated while still maintaining a reasonable level of accuracy. This dimensionality reduction can lead to increased learning accuracy and improving comprehensibility [172], in addition to reducing inference time, which is a key element in practical applications [196].

The inference time can be tied to the size of the model [131] and as a reduction in features leads to a significant decrease in model size [177], [181], it can also lead to a reduction in inference time. Feature reduction strategies, such as those enabled by EFR-TMD, are critical in addressing the rising energy demands of machine learning models, as prior studies have shown that reducing dataset size and complexity can lead to a 23–99% decrease in energy consumption and carbon emissions without compromising accuracy [174]. Furthermore, there are potential memory savings when removing features from neural networks [177]. Note that a reduction of features does not always lead to smaller models. The model size of long short-term memories is mainly dependent on its architecture and not the number of features. Similarly, the model size of tree-based algorithms is generally unaffected [177]. However, tree-based algorithms, such as Random Forest and XGBoost, may increase in size when subjected to feature reduction. This observation is in line with prior findings that suggest adopting simpler models over complex, highly parameterised models can significantly reduce energy consumption while maintaining comparable accuracy, reinforcing the importance of structured feature selection in sustainable machine learning practices [174]. Tree-based algorithms tend to compensate for the lack of discriminative power by creating a more complex tree structure. As the size of the tree-based algorithms are increasing as features are removed one could argue that the features being removed contains important information that is useful to discern the mode of transportation. While features deemed less important could still contribute to the classification, this could be a result of the initial benchmark model being overfitted and by reducing the number of features the model must instead focus on different, more applicable patterns.

Feature reduction also contributes to significantly decrease training time of up to 80%, which frees up computational resources, speeding up analysis and development, and in turn contribute to energy savings [174], [180]. Beyond performance metrics, adopting responsible machine learning practices is crucial in transport mode detection, particularly in energy-constrained environments such as mobile devices. Studies indicate that the energy footprint of machine learning is often overlooked, yet conscious selection of features and models can reduce unnecessary computational overhead, contributing to more sustainable deployment [174].

5.3.4 Limitations

While EFR-TMD have demonstrated several benefits, it is worth noting its limitations as well. First of all, while the framework has been evaluated using the NOR-TMD dataset, its performance on datasets with differing characteristics has yet to be investigated. While EFR-TMD was tested across a wide array of commonly used algorithms, it does not encompass all possible models, raising the possibility that results may vary when applied to substantially different algorithms.

Another limitation is the scope of features considered during the identification of applicable sensors and aggregation functions for transport mode detection. While data from all common smartphone sensors was considered, the evaluation included only features derived from the time domain. Had frequency domain features been incorporated, the results might have differed. That being said, transforming features to the frequency domain is highly energy consuming [128], which in turn reduce the applicability of the solution for resource-constrained devices.

Another aspect is the sizes of the windows used for segregating the data. Windows with 10-second windows with a 5-second overlap were employed while evaluating EFR-TMD. While this was well-founded in previous research [14], [73], [121], [191], during consecutive experiments it was discovered that reducing the overlap of the segments further improved the accuracy and robustness of the models. As such, a different configurations could potentially had influenced evaluation results of EFR-TMD.

Furthermore, advanced processing techniques, such as smoothing, were not applied. Data derived from different sensors often require tailored processing methods, however the goal of this evaluation was to assess all features in a generic and consistent manner. Consequently, sensor-specific preprocessing was not employed and introducing appropriate processing techniques for each sensor type could potentially alter the observed importance of individual sensors and provide further insights into their relative contributions.

Despite room for improvement, this thesis have presented a standardised framework for feature evaluation and reduction, able to systematically and consistently identify relevant features across multiple machine learning algorithms. This in turn enables efficient dimensionality reduction, improving computational efficiency.

5.4 Platform-Agnostic Framework for Local Transport Mode Detection

The main research question posed in this thesis was to investigate how an efficient, on-device, and platform-agnostic transport mode detection framework can be developed for real-time inference on mobile devices. Building on insights derived from the evaluation of smartphone sensors, aggregation functions, and iterative experimentation with various algorithmic approaches (Articles II-IV), this research presents an instantiation of a platform-agnostic transport mode detection framework (Article V), evaluated within the DSR paradigm. To the best of current knowledge, the only comparable solution is EdgeTrans [56], which employs a tree-based classifier to classify transportation modes across both Android and iOS, in contrast to the neural network utilised in this research. A platform-agnostic approach offers several benefits. First, it enables the use of the same model across multiple platforms, meaning only one model needs to be trained for applications intended to support various platforms. Since training machine learning models requires substantial computational resources and energy expenditure [180], eliminating the need for multiple models can yield significant energy savings. Furthermore, employing a single model ensures consistent behaviour across devices and operating systems. Evaluation results using a holdout set of unseen data (Table 4.11) demonstrate consistent performance and reasonable accuracy across both operating systems.

5.4.1 On-Device Inference

The fact that the platform-agnostic framework operates fully on-device also comes with several benefits compared to studies based on remote architectures. A major concern with remote transport mode detection is that sensitive data is stored centrally. While this concern has traditionally been mostly related to location data [66], it also applies to inertial sensor data [67]. Although less precise than location data, accelerometer data can still be used to identify individuals, determine their home and work locations, and even infer religious or political affiliations [67]. As such, a key advantage of local transport mode detection is that no sensitive data needs to leave the device.

Moreover, as data does not need to travel to a centralised solution, there is no added latency beyond that imposed by the window size and inference time [65]. Since Wi-Fi is also highly energy-consuming [65], eliminating continuous data transfer further contributes to overall energy savings. Additionally, centralised solutions must maintain a network connection for data transmission and receiving predictions, which can be challenging in underground public transport systems, where connectivity is often unreliable [202]. Furthermore, evaluations in this thesis, as well as in previous works [14], [56], indicate that local transport mode detection can be performed efficiently without significant energy consumption. Thus, achieving high-performing local transport mode detection offers a more privacy-preserving alternative to centralised approaches while also reducing latency, ultimately improving user experience.

5.4.2 Evaluation of Performance, Efficiency, and Energy Consumption

The framework was evaluated across multiple devices and operating systems, reinforcing its applicability in real-life Intelligent Transportation Systems while ensuring compliance with key constraints such as energy efficiency, inference time, and user privacy. The proposed solution demonstrates significantly faster inference times compared to other approaches. While Ferreira, Zavgorodnii, and Veiga [56] does not specify their model's inference time, the proposed system requires between 30 to 120 seconds in order to make a classification. The platform-agnostic framework proposed in this thesis requires only 15 seconds, in addition to the inference time required by the model to classify the processed data. As such, the platform-agnostic framework introduced in this research is able to make a classification with similar performance between 100% and 700% faster than previous work [56]. Compared to other studies, which report inference times ranging from 32 ms to 292 ms [14], this framework achieves an average inference time of 5.31 ms on Android and 2.05 ms on iOS. A significant variation in inference time was observed across devices (Figure 4.17), which can be attributed to differences in hardware and software. However, a notable relationship was also observed between inference time and problematic modalities, with train classification exhibiting both the longest inference time and the greatest classification difficulty.

Another advantage of the proposed framework is its significantly reduced model size. According to EFR-TMD results (Tables 4.9 and 4.10), a multilayer perceptron results in a model size that is over 5000 times smaller than a random forest classifier, utilised by Ferreira, Zavgorodnii, and Veiga [56], making it more suitable for resource-constrained devices. Despite the reduced model complexity, the framework maintains competitive classification accuracy and supports a wider range of public transportation modalities compared to Ferreira, Zavgorodnii, and Veiga [56]. Additionally, the reduced window size of 15 seconds improves inference time, resource consumption, and model efficiency [73], making the framework particularly well-suited for real-time applications.

Similar to previous work [14], [56], [64], the energy consumption of Android devices running the proposed framework was assessed. Using the Android Debug Bridge (ADB), energy consumption was estimated to be between 0.83% and 2.79% per hour. These measurements align with the reported energy consumption in Ferreira, Zavgorodnii, and Veiga [56] and Kamalian and Ferreira [64]. However, unlike prior studies, the energy consumption estimates in this research include the screen being continuously on. Given that screen usage accounts for most of the energy consumed, the proposed framework likely offers an even more energy-efficient solution than that of Ferreira, Zavgorodnii, and Veiga [56]. However, these remain rough estimates, as battery behaviour is influenced by factors such as temperature and system load [204]. Interestingly, a correlation was observed between inference time and energy expenditure, where the device with the fastest inference time also exhibited the highest energy consumption. Conversely, the device with the slowest inference time had the lowest energy usage, suggesting that inference speed directly impacts power consumption.

5.4.3 Limitations

Despite the advantages of the lightweight platform-agnostic framework, certain limitations must be acknowledged. The real-life assessment reveals less consistent performance for specific transport modes, particularly on Android devices, which underperform compared to iOS devices in real-life settings. Several factors may contribute to this discrepancy. Notably, the participants in the real-life evaluation were not part of the original data collection, which was an intentional design choice to assess generalisation to new users. However, variations in how users interact with their devices can influence sensor readings [61], [143]. While participants were instructed to hold their devices consistently, the sensitivity of smartphone sensors [85], [135] and individual user habits [188] may have introduced variability not present in the training data. If user behaviour significantly affected sensor readings, one would expect similar discrepancies on iOS devices, which were not observed. A more likely explanation is the greater consistency in hardware and sensor placement in iOS devices, as opposed to Android devices, which exhibit significant variability across manufacturers. Among the tested devices (Table 3.3), only one of the devices employed in the evaluation was included in the original data collection, and the lack of data from other devices could contribute to Android's suboptimal performance.

Furthermore, classification accuracy for train and tram modes was lower across both operating systems. This performance gap is likely due to the limited amount of training data available for these specific modes, as indicated by the strong correlation between classifier performance and dataset size (Article V). Previous studies incorporating rail-based transport modes have also reported suboptimal results, even in controlled simulated environments [60], [126], [131], [143]. Many studies have responded to this challenge by excluding certain rail-based modes [74], [124], [127], [128], [130], [134], [140] or by reducing modal granularity, such as grouping all rail-based transportation [94] or even all motorised transportation [129]. Additionally, while reducing the feature set and window size improves efficiency, it may also contribute to the decreased performance observed for specific transport modes. Although discrepancies exist between simulated and real-life settings, the proposed framework presents a scalable, privacy-conscious, and energy-efficient alternative for on-device transport mode detection.

Chapter 6

Conclusion

This thesis systematically investigated key challenges in transport mode detection (TMD) through the lens of the Design Science Research (DSR) paradigm, leading to the development of multiple artefacts that addressed critical limitations within the field. The overarching research question of this thesis explored how efficient, on-device, platform-agnostic transport mode detection could be achieved on mobile devices. Addressing this objective involved multiple considerations, which led to three supporting research questions.

The first supporting question focused on predictive accuracy, specifically examining the challenge of achieving high accuracy across diverse modes of transportation. This issue was investigated iteratively through multiple cycles of design and refinement of machine learning models. While algorithm selection and configuration played a crucial role in predictive performance, a diverse dataset and extracted features proved to be the decisive factors in improving accuracy across a broad spectrum of transportation modes. However, no standardised methodologies existed for identifying and extracting the most relevant features.

Consequently, the second supporting research question investigated how a standardised framework for feature evaluation and reduction could systematically identify relevant features by determining the most informative sensors. Consistent with previous findings, singular feature estimation techniques failed to reliably identify important features across different algorithms when applied to the same dataset. To address this limitation, this thesis proposed an ensemble feature-ranking approach. This approach was evaluated across a broad range of machine learning algorithms applicable to transport mode detection, demonstrating consistent results. The findings suggested that the absence of standardised feature evaluation methodologies could be mitigated through ensembles of existing techniques.

As the overarching research question encompassed on-device, platform-agnostic transport mode detection, the third supporting research question examined how machine learning models could be optimised for real-time, low-latency inference while maintaining computational efficiency on mobile devices. This question was addressed through investigations into feature reduction in combination with real-world evaluations. By identifying the most relevant features and eliminating redundant or irrelevant ones, both model size and inference time were significantly reduced, facilitating deployment on mobile devices. The research activities conducted in this thesis considered both Android and iOS platforms, individually and collectively, to develop platform-agnostic models. The solution to the platform-agnostic component of the overarching research question lay in normalising sensor data from both platforms and ensuring that data processing accounted for platform-specific differences. Through real-world testing, this thesis demonstrated that platform-agnostic transport mode detection on mobile devices was feasible in an efficient manner.

6.1 Summary of Research Questions and Key Findings

This thesis set out to investigate efficient, on-device, platform-agnostic transport mode detection using mobile devices. To achieve this, one overarching research question and three supporting research questions were formulated (RQ0–RQ3). Below, each research question is revisited and answered based on the key findings.

RQ0: How can efficient on-device, platform-agnostic transport mode detection be achieved on mobile devices?

Answer: Efficient and platform-agnostic transport mode detection is achieved through an integrated approach combining model optimisation (RQ3), robust generalisation across devices and users (RQ1), and systematic feature reduction (RQ2). By developing models and evaluation pipelines that operate effectively across Android and iOS devices, and by demonstrating low-latency on-device performance with minimal energy overhead, the thesis provides a cohesive solution. The contributions collectively show that centralised cloud-based architectures can be replaced with efficient, accurate, and real-time on-device alternatives.

RQ1: How can machine learning models achieve high accuracy in transport mode detection across diverse transport modes, ensuring generalisability in real-life applications?

Answer: High accuracy in transport mode detection was achieved using algorithms such as XGBoost and multilayer perceptrons trained on the NOR-TMD dataset, which includes real-world sensor data from over 100 participants. These models showed strong performance on unseen data across diverse transport modes and platforms. While real-life deployment of the MLP model showed a drop in accuracy compared to holdout tests, it maintained consistent performance across both Android and iOS platforms. This confirms that generalisable and robust transport mode detection is feasible, even under the variability of real-world conditions.

RQ2: How can a standardised framework for feature evaluation and reduction systematically identify relevant features, ensuring consistency and enabling reliable feature reduction across machine learning models?

Answer: The EFR-TMD framework enables consistent and reliable feature reduction through a model-agnostic approach that integrates multiple feature importance techniques. By normalising and aggregating importance scores across these methods, the framework identifies features that are consistently ranked as relevant, reducing bias from any single technique. EFR-TMD was validated across both Android and iOS sensor datasets, where key features, such as accelerometer and pressure sensor data, emerged as consistently important. Its reliability was further confirmed by applying it to five diverse classifiers. Across models, removing up to 50% of the lowest-ranked features had minimal impact on accuracy, while improving inference speed and reducing model size. This confirms that EFR-TMD effectively preserves informative features while discarding redundant ones, making it a practical and transferable tool for efficient model development on mobile platforms.

RQ3: How can transport mode detection models be optimised for real-time, low-latency inference while maintaining computational efficiency on mobile devices?

Answer: Transport mode detection models can be optimised for real-time, low-latency inference on mobile devices through a combination of dimensionality reduction, lightweight model architectures, and platform-agnostic implementation strategies. This research demonstrated that a reduced feature set, identified via the EFR-TMD framework, enabled a compact multilayer perceptron (MLP) model to run efficiently on both Android and iOS platforms. Real-world testing showed average inference times of 5.31 ms (Android) and 2.05 ms (iOS), with minimal misclassification between public and non-public transport modes. The model maintained low energy consumption (0.83%–2.79% per hour on Android), indicating suitability for continuous on-device use. These results confirm that real-time, computationally efficient transport mode detection is feasible on resource-constrained devices without relying on external servers, enabling scalable and privacy-preserving deployments.

Together, the answers to RQ1–RQ3 converge to answer the overarching research question (RQ0), demonstrating a practical, efficient, and generalisable framework for transport mode detection on mobile devices. This work lays a foundation for scalable deployment of intelligent transport systems leveraging only the sensors and computation available on users' smartphones.

6.2 Summary of Contributions

By iteratively designing, implementing, and evaluating solutions to address the aforementioned research questions, this thesis has made multiple significant contributions. These contributions have not only advanced theoretical understanding but have also resulted in tangible artefacts that can be utilised by researchers and practitioners to enhance transport mode detection (TMD) systems in real-world applications.

NOR-TMD: A Comprehensive Dataset for Transport Mode Detection

A central contribution of this thesis is the NOR-TMD dataset, which enhances the generalisability and applicability of TMD models by providing a comprehensive and diverse resource. Unlike previous datasets, which were limited in scope, device diversity, and sensor coverage, NOR-TMD incorporates data collected across multiple devices, operating systems, and travel contexts. Notably, it includes iOS sensor data, addressing a critical gap in the field, as previous datasets predominantly focused on Android-based implementations. By offering a more diverse dataset with extensive sensor coverage and device placement variations, NOR-TMD provides a robust foundation for developing accurate and widely applicable TMD models.

Accurate and Generalisable Transport Mode Classification

Building on this dataset, this thesis introduced high-accuracy transport mode detection models based on XGBoost, which demonstrated superior classification performance across diverse transport modes. The results highlight the effectiveness of gradient boosting approaches, which outperform traditional classifiers while maintaining efficiency across different device configurations. However, due to their computational complexity, gradient-boosted models are better suited for centralised implementations, which, despite their scalability, introduce privacy concerns related to continuous data transmission.

EFR-TMD: A Framework for Feature Evaluation and Reduction

To further standardise feature selection methodologies, this thesis introduced the EFR-TMD framework, an ensemble-based feature ranking method that systematically evaluates and reduces feature sets for TMD models. By aggregating multiple feature selection techniques, EFR-TMD provides a more reliable and generalisable approach to identifying the most relevant features for transport mode classification. The framework demonstrated that significant reductions in feature dimensionality does not compromise classification performance, leading to greater computational efficiency and lower energy consumption. Additionally, by confirming the critical role of sensors such as the accelerometer, magnetometer, and barometer, and ensuring cross-platform consistency, EFR-TMD supports the development of platform-agnostic TMD solutions.

Lightweight, Platform-Agnostic On-Device Framework

A major practical outcome of this thesis is the development of a lightweight, platform-agnostic transport mode detection framework designed for on-device inference. Unlike centralised architectures, which rely on continuous data transmission, this framework operates entirely on the user's device, eliminating privacy concerns and reducing energy consumption. Evaluation results confirmed that this approach maintains competitive classification performance while significantly improving inference time, making it well suited for real-time applications in Intelligent Transportation Systems. By leveraging insights from feature importance analysis, the framework selects only the most relevant features, further enhancing efficiency and scalability. While challenges remain, particularly in improving classification performance for trains and trams, the proposed framework represents a substantial advancement in deployable, privacy-preserving TMD solutions.

6.3 Summary of Limitations

This section provides a summary of the main limitations of the research. It focuses on challenges related to the dataset, which forms the foundation of all models and frameworks developed in this work, as well as limitations specific to the two frameworks presented in the thesis.

Limitations of the Proposed Dataset

While the NOR-TMD dataset constitutes a valuable and diverse resource for transport mode detection, it presents several limitations. A key issue is the imbalance across users, devices, and transport modes, which can introduce bias and increase the risk of overfitting. The labelling scheme, particularly the inside and outside categorisation, may lack the granularity needed for fine-grained behaviour analysis. More detailed activity labels could enhance the dataset's utility and compatibility with other datasets. Another limitation relates to the irregular sampling frequency of sensor data. Sensor events were captured using operating system triggers based on change, leading to inconsistent sampling intervals. This variability can negatively impact model performance and complicates generalisation across scenarios.

Limitations of the Proposed Frameworks

The EFR-TMD framework was evaluated on the NOR-TMD dataset but has not yet been validated on datasets with differing characteristics. Its generalisability to datasets with different user behaviours, devices, or sensor configurations has not been assessed. Only time-domain features were considered in the feature selection process. Frequency-domain features, though potentially valuable, were excluded due to their computational and energy demands, which limits applicability in resource-constrained environments. The choice of window size and overlap used during evaluation, while grounded in prior work, may have influenced the results. It was later discovered that reducing the segment overlap improved model robustness, suggesting that alternative configurations could yield different outcomes. Additionally, no sensor-specific preprocessing techniques were applied. The decision to evaluate features in a uniform manner provided consistency but may have obscured the relative importance of features derived from different sensors.

The platform-agnostic framework evaluated in this thesis demonstrated promising results. However, real-life testing revealed performance inconsistencies, especially on Android devices. This discrepancy is likely due to hardware heterogeneity and sensor placement variability across manufacturers. In contrast, iOS devices displayed more consistent performance, likely due to greater hardware standardisation. Furthermore, the participants in the real-life evaluation differed from those in the data collection phase, which was intended to test generalisation. Nevertheless, this introduced variability in user-device interaction that may have affected classification accuracy. Lower classification accuracy was also observed for rail-based transport modes, attributed to the limited volume of training data for these categories. This trend aligns with prior studies and indicates a need for more comprehensive data collection. Finally, while the framework prioritised efficiency through reduced feature sets and smaller window sizes, these design choices may have impacted performance for certain transport modes.

6.4 Theoretical Implications

This research contributes to the field of transport mode detection (TMD) by advancing the theoretical understanding of machine learning applications in mobility research. A key theoretical contribution is the demonstration of extreme gradient boosting (XGBoost) as a superior classification method for transport mode inference, reinforcing prior evidence that tree-based algorithms outperform other traditional approaches such as random forest, decision trees, and support vector machines. The superior performance of the XGBoost model trained on platform-specific data compared to other models, including cross-platform approaches employing different algorithms, suggests that tailoring models or preprocessing pipelines to specific platform characteristics may help reduce variability caused by heterogeneous hardware and software configurations. However, due to differences in model architectures, direct comparisons between platform-specific and cross-platform models remain challenging. This indicates the potential for future research to explicitly consider platform-dependent sensor variability to improve the generalisability and real-world applicability of transport mode detection models.

Furthermore, this research proposes a structured approach to feature evaluation and reduction (EFR-TMD), addressing a long-standing gap in standardising feature selection methods within TMD. By integrating multiple feature selection techniques into an ensemble-based framework, EFR-TMD enhances reproducibility and generalisability across different machine learning models. This addresses the common practice of selecting features based on intuition or arbitrary experimentation, providing a systematic methodology for future studies to identify relevant sensors and extraction techniques. The results indicate that up to 75% of commonly used features can be eliminated without significantly degrading classification performance, highlighting the potential for optimising feature selection in mobility-related research.

Another important theoretical implication is the role of dataset diversity in transport mode classification. The introduction of the NOR-TMD dataset extends the range of transportation modes studied in previous research and incorporates sensor data from both Android and iOS devices, an aspect rarely explored in existing datasets. In addition, the dataset provides a larger diversity in terms of devices and sensors, compared to existing datasets. This thesis demonstrates that sensor placement, participant diversity, and device heterogeneity can play crucial roles in model robustness, providing a foundation for future research on dataset representativeness in transport mode detection. Notably, models trained on the proposed NOR-TMD dataset outperform many existing alternatives, suggesting that the dataset's diversity may be a key factor contributing to improved generalisability.

Finally, this research contributes to the theoretical understanding of how accuracy, computational efficiency, and platform compatibility can be addressed simultaneously in transport mode detection. It demonstrates that a single, lightweight model can perform consistently across mobile platforms when supported by structured feature selection and dimensionality reduction. This finding expands the theoretical perspective on mobile sensing and embedded machine learning by highlighting how streamlined model architectures can support generalisable behaviour across heterogeneous device environments. The research also underscores the value of integrating feature selection into the model design process, showing that reduced input complexity can enhance efficiency while preserving classification accuracy. These insights contribute to a more nuanced theoretical understanding of how cross-platform robustness can be achieved through systematic feature selection, preprocessing, and model design.

6.5 Practical Implications

The findings of this research have significant practical implications for the deployment and scalability of transport mode detection (TMD) systems in real-life applications. The development of high-accuracy machine learning models for transport mode detection, capable of distinguishing between a broad range of transportation modes, has direct applications in Smart Mobility, public transportation systems, and automated ticketing solutions. The demonstrated ability to classify transport modes with over 98% accuracy suggests that Intelligent Transportation Systems (ITS) can leverage such models to improve public transport monitoring, optimise urban mobility infrastructure, and enhance passenger experiences through real-time travel insights.

The platform-agnostic, on-device TMD framework developed in this research introduces a practical alternative to cloud-based solutions, enabling privacy-preserving, real-time inference on mobile devices. This is particularly relevant for automated fare collection (AFC) systems, where transport mode detection can complement traditional ticketing mechanisms by providing an additional layer of validation for passenger movement. The framework offers a cost-effective and scalable approach for transit operators seeking to enhance fare validation systems, with the potential for integration into broader mobility solutions.

Additionally, the research highlights the energy efficiency benefits of feature selection and model optimisation. By demonstrating that EFR-TMD-based feature reduction can improve computational efficiency without sacrificing accuracy, this thesis provides actionable insights for developers designing real-time transport mode detection applications. The reduced computational overhead translates into lower energy consumption, making the proposed framework particularly suitable for resource-constrained mobile devices. Moreover, the research highlights how dimensionality reduction can reduce inference times, which can have practical implications for real-life solutions.

Finally, from a policy and urban planning perspective, the findings provide a valuable foundation for transport authorities and policymakers to enhance public transport planning and infrastructure development. The ability to accurately track and analyse travel patterns can support evidence-based decision-making on issues such as route optimization, service frequency adjustments, and urban mobility policies. By integrating transport mode detection into real-time public transport analytics, city planners can develop data-driven solutions to improve congestion management, reduce emissions, and enhance multimodal transport systems.

6.6 Directions for Future Research

While this research has made significant strides in advancing transport mode detection, it also identifies several areas for future exploration. Although the dataset introduced in this thesis addresses many of the limitations present in existing datasets, the inherent nature of machine learning dictates that an abundance of diverse and high-quality data is always beneficial. Simultaneously, the use of disparate datasets across studies complicates direct comparisons between proposed solutions, hindering the development of standardised benchmarks. As such, future efforts should focus on expanding the NOR-TMD dataset and integrating existing datasets to create a comprehensive and standardised benchmark dataset. Such an initiative would facilitate more robust comparisons across different methodologies and contribute to the overall advancement of transport mode detection research.

The framework for feature evaluation and reduction (EFR-TMD), introduced in this thesis, should be further assessed using additional feature types, particularly frequency-domain features, to rigorously evaluate its robustness. Similarly, the impact of segment size selection on the performance of the framework warrants further investigation to determine optimal configurations. While the ensemble-based approach integrates several commonly used feature importance techniques, additional research should explore whether alternative techniques should be incorporated to increase the robustness of the framework. Furthermore, the applicability of the framework should be assessed in broader contexts, such as human activity recognition, to determine its generalisability across various domains beyond transport mode detection.

The platform-agnostic framework developed in this research should be further refined to address classification challenges related to similar transport modalities. While the framework demonstrates high energy efficiency and consistency across both Android and iOS devices, there is clear potential for accuracy improvements, particularly in distinguishing closely related transport modes. Additionally, expanding the framework to accommodate a wider range of transport modalities would enhance its practical utility for public transport operators and other stakeholders. In a real-life deployment scenario, particularly in fully automated fare collection systems, the framework would need to continuously infer the mode of transportation, seamlessly detecting when a traveller enters or exits a specific mode. However, continuous inference is likely to be energy-intensive, necessitating the development of efficient trigger mechanisms to detect potential transitions between transport modes. Future research should explore optimal strategies for identifying and leveraging such trigger events, enabling a more resource-efficient transport mode detection process.

Ultimately, the research presented in this thesis provides a solid foundation for further advancements in transport mode detection, emphasising the importance of diversity, methodological rigour, and practical applicability in the development of intelligent transportation solutions. By addressing persistent challenges related to data availability,

Kristiania

feature selection, computational efficiency, and platform compatibility, this thesis contributes to the broader goal of creating scalable, privacy-conscious, and high-performance transport mode detection systems that can support the next generation of Smart Mobility applications.

Bibliography

[1] U. Nations, *68 of the world population projected to live in urban areas by 2050*. [Online]. Available: `https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html`.

[2] S. Angel, J. Parent, D. L. Civco, A. Blei, and D. Potere, "The dimensions of global urban expansion: Estimates and projections for all countries, 2000–2050," *Progress in Planning*, The dimensions of global urban expansion: Estimates and projections for all countries, 2000–2050, vol. 75, no. 2, pp. 53–107, Feb. 1, 2011.

[3] C. N. N. Tran, T. T. H. Tat, V. W. Y. Tam, and D. H. Tran, "Factors affecting intelligent transport systems towards a smart city: A critical review," *International Journal of Construction Management*, vol. 23, no. 12, pp. 1982–1998, Sep. 10, 2023.

[4] P. Yu, Y. Wei, L. Ma, B. Wang, E. H. K. Yung, and Y. Chen, "Urbanization and the urban critical zone," *Earth Critical Zone*, vol. 1, no. 1, p. 100 011, Jun. 1, 2024.

[5] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustainable Cities and Society*, vol. 38, pp. 697–713, Apr. 1, 2018.

[6] R. Mangiaracina, A. Perego, G. Salvadori, and A. Tumino, "A comprehensive view of intelligent transport systems for urban smart mobility," *International Journal of Logistics Research and Applications*, vol. 20, no. 1, pp. 39–52, Jan. 2, 2017.

[7] A. Camero and E. Alba, "Smart city and information technology: A review," *Cities*, vol. 93, pp. 84–94, Oct. 1, 2019.

[8] M. A. Hassan, R. Javed, Farhatullah, *et al.*, "Intelligent transportation systems in smart city: A systematic survey," in *2023 International Conference on Robotics and Automation in Industry (ICRAI)*, Mar. 2023, pp. 1–9.

[9] R. Weiland and L. Purser, "Weiland: Intelligent transportation systems - Google Scholar," *Committee on Intelligent Transportation Systems*, 2000.

[10] X. Basagaña, M. Triguero-Mas, D. Agis, *et al.*, "Effect of public transport strikes on air pollution levels in barcelona (spain)," *Science of The Total Environment*, vol. 610-611, pp. 1076–1082, Jan. 1, 2018.

[11] D. Q. Nguyen-Phuoc, G. Currie, C. De Gruyter, and W. Young, "Exploring the impact of public transport strikes on travel behavior and traffic congestion," *International Journal of Sustainable Transportation*, vol. 12, no. 8, pp. 613–623, Sep. 14, 2018.

[12] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[13] A. C. Prelipcean, G. Gidófalvi, and Y. O. Susilo, "Transportation mode detection – an in-depth review of applicability and reliability," *Transport Reviews*, vol. 37, no. 4, pp. 442–464, Jul. 4, 2017.

[14] M. Oplenskedal, A. Taherkordi, and P. Herrmann, "ATARAXIS: A deep learning approach for hardwareless in-vehicle presence detection," in *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*, Dec. 2021, pp. 01–10.

[15] L. Figueiredo, I. Jesus, J. Machado, J. Ferreira, and J. Martins de Carvalho, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, Aug. 2001, pp. 1206–1211.

[16] K. N. Qureshi and A. H. Abdullah, "A survey on intelligent transportation systems," *Middle-East Journal of Scientific Research*, vol. 15, no. 5, pp. 629–642, 2013.

[17] S. Kaffash, A. T. Nguyen, and J. Zhu, "Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis," *International Journal of Production Economics*, vol. 231, p. 107 868, Jan. 2021.

[18] S.-h. An, B.-H. Lee, and D.-R. Shin, "A survey of intelligent transportation systems," in *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*, Jul. 2011, pp. 332–337.

[19] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3152–3168, Aug. 2020.

[20] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[21] R. Faria, L. Brito, K. Baras, and J. Silva, "Smart mobility: A survey," in *2017 International Conference on Internet of Things for the Global Community (IoTGC)*, Jul. 2017, pp. 1–8.

[22] P. A. Maldonado Silveira Alonso Munhoz, F. da Costa Dias, C. Kowal Chinelli, *et al.*, "Smart mobility: The main drivers for increasing the intelligence of urban mobility," *Sustainability*, vol. 12, no. 24, p. 10 675, Jan. 2020.

[23] S. Paiva, M. A. Ahad, G. Tripathi, N. Feroz, and G. Casalino, "Enabling Technologies for Urban Smart Mobility: Recent Trends, Opportunities and Challenges," en, *Sensors*, vol. 21, no. 6, p. 2143, Jan. 2021.

[24] N. B. Aletà, C. M. Alonso, and R. M. A. Ruiz, "Smart mobility and smart environment in the spanish cities," *Transportation Research Procedia*, 3rd Conference on Sustainable Urban Mobility, 3rd CSUM 2016, 26 – 27 May 2016, Volos, Greece, vol. 24, pp. 163–170, Jan. 1, 2017.

[25] M. Trépanier, N. Tranchant, and R. Chapleau, "Individual trip destination estimation in a transit smart card automated fare collection system," *Journal of Intelligent Transportation Systems*, vol. 11, no. 1, pp. 1–14, Apr. 1, 2007.

[26] K. Witvoet, C. Vidal, T. Stiene, and A. Emadi, "Towards frictionless public transit: A brief review of automatic fare collection," in *2024 IEEE International Conference on Smart Mobility (SM)*, Sep. 2024, pp. 87–92.

[27] R. Risser, S. Iwarsson, and A. Ståhl, "How do people with cognitive functional limitations post-stroke manage the use of buses in local public transport?" *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 15, no. 2, pp. 111–118, Mar. 1, 2012.

[28] A. B. R. González, J. J. V. Díaz, and M. R. Wilby, "Detailed origin-destination matrices of bus passengers using radio frequency identification," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 1, pp. 141–152, Jan. 2022.

[29] T. Apanasevic and J. Markendahl, "Mobile ticketing services in the northern europe: Critical business model issues," in *2017 Internet of Things Business Models, Users, and Networks*, Nov. 2017, pp. 1–8.

[30] M. A. Bitew, A. Muhammad, D. H. Fandiantoro, H. Boedinoegroho, and A. Kurniawan, "E-payment for public transportation using BIBO method based on bluetooth low energy beacon," in *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, Nov. 2020, pp. 199–204.

[31] L. Hasimi, A. Poniszewska-Marańda, and T. Krym, "Smart city concept using BI/BO model for the ticket system in urban transport," in *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2021, pp. 1–6.

[32] G. Tuveri, M. Garau, E. Sottile, *et al.*, "Automating ticket validation: A key strategy for fare clearing and service planning," in *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Jun. 2019, pp. 1–10.

[33] T.-M. Grønli, A. Lakhan, and S. Memon, "A novel BIBO automated ticketing system based on blockchain mobile sensors for public transport modes," in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, Jun. 2024, pp. 01–05.

[34] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselböck, and N. Höfler, "Be-In/Be-Out with Bluetooth Low Energy: Implicit Ticketing for Public Transportation Systems," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 1551–1556.

[35] B. Wieczorek and A. Poniszewska-Marańda, "Be in/Be out model for intelligent transport in SmartCity approach," in *Proceedings of the 17th International Conference on Advances in Mobile Computing & Multimedia*, Munich Germany: ACM, Dec. 2, 2019, pp. 226–230.

[36] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselböck, and N. Höfler, "Bluetooth Low Energy as enabling technology for Be-In/Be-Out systems," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2016, pp. 423–428.

[37] M. W. Mastalerz, A. Malinowski, S. Kwiatkowski, A. Śniegula, and B. Wieczorek, "Passenger BIBO detection with IoT support and machine learning techniques for intelligent transport systems," *Procedia Computer Science*, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020, vol. 176, pp. 3780–3793, Jan. 1, 2020.

[38] M. Bieler, A. Skretting, P. Büdinger, and T.-M. Grønli, "Survey of Automated Fare Collection Solutions in Public Transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 248–14 266, Sep. 2022.

[39] R. Couto, J. Leal, P. M. Costa, and T. Galvão, "Exploring ticketing approaches using mobile technologies: QR codes, NFC and BLE," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 7–12.

[40] V. Kostakos, T. Camacho, and C. Mantero, "Wireless detection of end-to-end passenger trips on public transport buses," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 1795–1800.

[41] V. Servizi, D. R. Persson, F. C. Pereira, *et al.*, "Large scale passenger detection with smartphone/bus implicit interaction and multisensory unsupervised cause-effect learning," *arXiv:2202.11962 [cs]*, Feb. 24, 2022. arXiv: `2202.11962`.

[42] V. Servizi., D. R. Persson, F. C. Pereira, *et al.*, ""is not the truth the truth?": Analyzing the impact of user validations for bus in/out detection in smartphone-based surveys," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 11 905–11 920, Nov. 2023. arXiv: `2202.11961[cs]`.

[43] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[44] D. Cianciulli, G. Canfora, and E. Zimeo, "Beacon-based context-aware architecture for crowd sensing public transportation scheduling and user habits," *Procedia Computer Science*, 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal, vol. 109, pp. 1110–1115, Jan. 1, 2017.

[45] J. Wahlström, I. Skog, and P. Händel, "Smartphone-based vehicle telematics: A ten-year anniversary," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2802–2825, Oct. 2017.

[46] K. E. Zannat and C. F. Choudhury, "Emerging big data sources for public transport planning: A systematic review on current state of art and future research directions," *Journal of the Indian Institute of Science*, vol. 99, no. 4, pp. 601–619, Dec. 1, 2019.

[47] R. Meng, D. W. Grömling, R. R. Choudhury, and S. Nelakuditi, "RideSense: Towards ticketless transportation," in *2016 IEEE Vehicular Networking Conference (VNC)*, Dec. 2016, pp. 1–8.

[48] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, Sep. 2010.

[49] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402–427, 2013.

[50] M. Grossi, "A sensor-centric survey on the development of smartphone measurement and sensing systems," *Measurement*, vol. 135, pp. 572–592, Mar. 1, 2019.

[51] W. Elmenreich, "An introduction to sensor fusion," *Vienna University of Technology, Austria*, vol. 502, pp. 1–28, 2002.

[52] P. Sadeghian, J. Håkansson, and X. Zhao, "Review and evaluation of methods in transport mode detection based on GPS tracking data," en, *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 8, no. 4, pp. 467–482, Aug. 2021.

[53] L. Shen and P. R. Stopher, "Review of GPS travel survey and GPS data-processing methods," *Transport Reviews*, vol. 34, no. 3, pp. 316–334, May 4, 2014.

[54] G. Xiao, Q. Cheng, and C. Zhang, "Detecting travel modes using rule-based classification system and Gaussian process classifier," *IEEE Access*, vol. 7, pp. 116 741–116 752, 2019.

[55] M. Oplenskedal, A. Taherkordi, and P. Herrmann, "DeepMatch: Deep matching for in-vehicle presence detection in transportation," in *Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems*, ser. DEBS '20, New York, NY, USA: Association for Computing Machinery, Jul. 13, 2020, pp. 97–108.

[56] P. Ferreira, C. Zavgorodnii, and L. Veiga, "edgeTrans - Edge transport mode detection," en, *Pervasive and Mobile Computing*, vol. 69, p. 101 268, Nov. 2020.

[57] P. Nirmal, I. Disanayaka, D. Haputhanthri, and A. Wijayasiri, "Transportation Mode Detection Using Crowdsourced Smartphone Data," in *2021 28th Conference of Open Innovations Association (FRUCT)*, Jan. 2021, pp. 341–349.

[58] T. Plötz and Y. Guan, "Deep Learning for Human Activity Recognition in Mobile Computing," *Computer*, vol. 51, no. 5, pp. 50–59, May 2018.

[59] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, 33:1–33:33, Jan. 1, 2014.

[60] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '13, New York, NY, USA: Association for Computing Machinery, Nov. 11, 2013, pp. 1–14.

[61] J. Biancat, C. Brighenti, and A. Brighenti, "Review of transportation mode detection techniques," *EAI Endorsed Transactions on Ambient Systems*, vol. "1", no. 4, Oct. 31, 2014.

[62] M. Nikolic and M. Bierlaire, "Review of transportation mode detection approaches based on smartphone data," in *17th Swiss Transport Research Conference*, 2017.

[63] C. Sarkar, J. J. Treurniet, S. Narayana, R. V. Prasad, and W. de Boer, "SEAT: Secure energy-efficient automated public transport ticketing system," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 222–233, Mar. 2018.

[64] M. Kamalian and P. Ferreira, "FogTMDetector - Fog Based Transport Mode Detection using Smartphones," in *2022 IEEE 6th International Conference on Fog and Edge Computing (ICFEC)*, May 2022, pp. 9–16.

[65] M. Kamalian, P. Ferreira, and E. Jul, "A survey on local transport mode detection on the edge of the network," en, *Applied Intelligence*, vol. 52, no. 14, pp. 16 021–16 050, Nov. 2022.

[66] Q. Huang, J. Zhang, Z. Zeng, D. He, X. Ye, and Y. Chen, "PPDF-FedTMI: A federated learning-based transport mode inference model with privacy-preserving data fusion," *Simulation Modelling Practice and Theory*, vol. 129, p. 102 845, Dec. 1, 2023.

[67] J. L. Kröger, P. Raschke, and T. R. Bhuiyan, "Privacy implications of accelerometer data: A review of possible inferences," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, ser. ICCSP '19, New York, NY, USA: Association for Computing Machinery, Jan. 19, 2019, pp. 81–87.

[68] J. Wang, B. Cao, P. Yu, L. Sun, W. Bao, and X. Zhu, "Deep learning towards mobile applications," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2018, pp. 1385–1393.

[69] X. Liang, Y. Zhang, G. Wang, and S. Xu, "A deep learning model for transportation mode detection based on smartphone sensing data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5223–5235, Dec. 2020.

[70] G. Asci and M. A. Guvensan, "A Novel Input Set for LSTM-Based Transport Mode Detection," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2019, pp. 107–112.

[71] D. Bachir, G. Khodabandelou, V. Gauthier, M. El Yacoubi, and E. Vachon, "Combining Bayesian Inference and Clustering for Transport Mode Detection from Sparse and Noisy Geolocation Data," en, in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Curry, E. Daly, *et al.*, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 569–584.

[72] C. Carpineti, V. Lomonaco, L. Bedogni, M. D. Felice, and L. Bononi, "Custom Dual Transportation Mode Detection By Smartphone Devices Exploiting Sensor Diversity," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2018, pp. 367–372.

[73] P. Matthes and T. Springer, "Selecting Resource-Efficient ML Models for Transport Mode Detection on Mobile Devices," in *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, Nov. 2022, pp. 135–141.

[74] B. Alotaibi, "Transportation Mode Detection by Embedded Sensors Based on Ensemble Learning," *IEEE Access*, vol. 8, pp. 145 552–145 563, 2020.

[75] T. H. Vu, L. Dung, and J.-C. Wang, "Transportation Mode Detection on Mobile Devices Using Recurrent Nets," in *Proceedings of the 24th ACM international conference on Multimedia*, ser. MM '16, New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 392–396.

[76] F. Asgari and S. Clemencon, "Transport Mode Detection when Fine-grained and Coarse-grained Data Meet," in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Sep. 2018, pp. 301–307.

[77] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and GIS information," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11, New York, NY, USA: Association for Computing Machinery, Nov. 1, 2011, pp. 54–63.

[78] Y.-J. Byon and S. Liang, "Real-time transportation mode detection using smartphones and artificial neural networks: Performance comparisons between smartphones and conventional global positioning system sensors," *Journal of Intelligent Transportation Systems*, vol. 18, no. 3, pp. 264–272, Jul. 3, 2014.

[79] B. Nham, K. Siangliulue, and S. Yeung, "Predicting mode of transport from iphone accelerometer data," *Machine Learning Final Projects, Stanford University*, 2008.

[80] A. Aziz, N. K. Youkee, F. S. Ahmed, S. Tasnim, M. Fahim-Ul-Islam, and A. Chakrabarty, "Enhancing transparency in transport mode detection: An interpretable ensemble model classifier," in *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*, Mar. 2024, pp. 1–6.

[81] M. Oplenskedal, P. Herrmann, and A. Taherkordi, "DeepMatch2: A comprehensive deep learning-based approach for in-vehicle presence detection," *Information Systems*, vol. 108, p. 101 927, Sep. 1, 2022.

[82] A. Jahangiri and H. A. Rakha, "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2406–2417, Oct. 2015.

[83] X. Zhu, J. Li, Z. Liu, S. Wang, and F. Yang, "Learning transportation annotated mobility profiles from GPS data for context-aware mobile services," in *2016 IEEE International Conference on Services Computing (SCC)*, Jun. 2016, pp. 475–482.

[84] M. M. Alam, T. Ahmed, M. Hossain, *et al.*, "Federated ensemble-learning for transport mode detection in vehicular edge network," *Future Generation Computer Systems*, vol. 149, pp. 89–104, Dec. 1, 2023.

[85] K.-Y. Chen, R. C. Shah, J. Huang, and L. Nachman, "Mago: Mode of transport inference using the hall-effect magnetic sensor and accelerometer," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 2, 8:1–8:23, Jun. 30, 2017.

[86] P. Nitsche, P. Widhalm, S. Breuss, N. Brändle, and P. Maurer, "Supporting large-scale travel surveys with smartphones – a practical approach," *Transportation Research Part C: Emerging Technologies*, Special Issue with Selected Papers from Transport Research Arena, vol. 43, pp. 212–221, Jun. 1, 2014.

[87] B. Assemi, H. Safi, M. Mesbah, and L. Ferreira, "Developing and validating a statistical model for travel mode identification on smartphones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1920–1931, Jul. 2016.

[88] I. Semanjski, S. Gautama, R. Ahas, and F. Witlox, "Spatial context mining approach for transport mode recognition from mobile sensed big data," *Computers, Environment and Urban Systems*, vol. 66, pp. 38–52, Nov. 1, 2017.

[89] F. Muharemi, E. Syka, and D. Logofatu, "Recognizing user's activity and transport mode detection: Maintaining low-power consumption," in *Machine Learning and Knowledge Discovery in Databases*, P. Cellier and K. Driessens, Eds., Cham: Springer International Publishing, 2020, pp. 21–37.

[90] E. Sadeghvaziri, M. B. Rojas, and X. Jin, "Exploring the potential of mobile phone data in travel pattern analysis," *Transportation Research Record*, vol. 2594, no. 1, pp. 27–34, Jan. 1, 2016.

[91] F. Biljecki, H. Ledoux, and P. van Oosterom, "Transportation mode-based segmentation and classification of movement trajectories," *International Journal of Geographical Information Science*, vol. 27, no. 2, pp. 385–407, Feb. 1, 2013.

[92] H. Dabbas and B. Friedrich, "Estimating traffic demand of different transportation modes using floating smartphone data," *Transportmetrica A: Transport Science*, vol. 0, no. 0, pp. 1–27,

[93] D. Bachir, G. Khodabandelou, V. Gauthier, M. El Yacoubi, and J. Puchinger, "Inferring dynamic origin-destination flows by transport mode using mobile phone data," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 254–275, Apr. 1, 2019.

[94] O. Lorintiu and A. Vassilev, "Transportation mode recognition based on smartphone embedded sensors for carbon footprint estimation," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 1976–1981.

[95] E.-H. Chung and A. Shalaby, "A trip reconstruction tool for GPS-based personal travel surveys," *Transportation Planning and Technology*, vol. 28, no. 5, pp. 381–401, Oct. 1, 2005.

[96] N. Cardoso, J. Madureira, and N. Pereira, "Smartphone-based transport mode detection for elderly care," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Sep. 2016, pp. 1–6.

[97] H. Wang, F. Calabrese, G. Di Lorenzo, and C. Ratti, "Transportation mode inference from anonymized and aggregated mobile phone call detail records," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 318–323.

[98] A. Bloch, R. Erdin, S. Meyer, T. Keller, and A. De Spindler, "Battery-efficient transportation mode detection on mobile devices," in *2015 16th IEEE International Conference on Mobile Data Management*, vol. 1, Jun. 2015, pp. 185–190.

[99] K. Chin, H. Huang, C. Horn, I. Kasanicky, and R. Weibel, "Inferring fine-grained transport modes from mobile phone cellular signaling data," *Computers, Environment and Urban Systems*, vol. 77, p. 101 348, Sep. 1, 2019.

[100] X. Gou, C.-C. Hung, G. Li, and W.-C. Peng, "PTGF: Public transport general framework for identifying transport modes based on cellular data," in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, Jun. 2019, pp. 563–568.

[101] H. Huang, Y. Cheng, and R. Weibel, "Transport mode detection based on mobile phone network data: A systematic review," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 297–312, Apr. 1, 2019.

[102] A. M. AbdelAziz, A. Shoukry, and M. Youssef, "Ubiquitous transportation mode detection using single cell tower information," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9250–9259, Apr. 2021.

[103] H.-K. Chen, H.-C. Ho, L.-Y. Wu, I. Lee, and H.-W. Chou, "Two-stage procedure for transportation mode detection based on sighting data," *Transportmetrica A: Transport Science*, vol. 20, no. 1, p. 2 118 558, Jan. 2, 2024.

[104] S. Mostafa, M. Youssef, and K. A. Harras, "ModeSense: Ubiquitous and accurate transportation mode detection using serving cell tower information," in *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '24, New York, NY, USA: Association for Computing Machinery, Nov. 22, 2024, pp. 184–195.

[105] Y.-J. Byon, B. Abdulhai, and A. Shalaby, "Real-time transportation mode detection via tracking global positioning system mobile devices," *Journal of Intelligent Transportation Systems*, vol. 13, no. 4, pp. 161–170, Nov. 6, 2009.

[106] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, "Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification," *Computers, Environment and Urban Systems*, Special Issue: Advances in Geocomputation, vol. 36, no. 6, pp. 526–537, Nov. 1, 2012.

[107] G. Xiao, Z. Juan, and C. Zhang, "Travel mode detection based on GPS track data and bayesian networks," *Computers, Environment and Urban Systems*, vol. 54, pp. 14–22, Nov. 1, 2015.

[108] S. Dabiri and K. Heaslip, "Inferring transportation modes from GPS trajectories using a convolutional neural network," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 360–371, Jan. 1, 2018.

[109] B. Wang, L. Gao, and Z. Juan, "Travel mode detection using GPS data and socioeconomic attributes based on a random forest classifier," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1547–1558, May 2018.

[110] B. Wang, Y. Wang, K. Qin, and Q. Xia, "Detecting transportation modes based on LightGBM classifier from GPS trajectory data," in *2018 26th International Conference on Geoinformatics*, Jun. 2018, pp. 1–7.

[111] S. Dabiri, C.-T. Lu, K. Heaslip, and C. K. Reddy, "Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 1010–1023, May 2020.

[112] L. Li, J. Zhu, H. Zhang, H. Tan, B. Du, and B. Ran, "Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using GPS data," *Transportation Research Part A: Policy and Practice*, vol. 136, pp. 282–292, Jun. 1, 2020.

[113] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Ensemble convolutional neural networks for mode inference in smartphone travel survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2232–2239, Jun. 2020.

[114] L. Gao, X. Chen, Z. Zhu, and T.-H. Chang, "Effectiveness of public transport networks in motorized mode detection: A case study of a planning survey in nanjing," in *2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*, Sep. 2020, pp. 37–43.

[115] J. Li, X. Pei, X. Wang, D. Yao, Y. Zhang, and Y. Yue, "Transportation mode identification with GPS trajectory data and GIS information," *Tsinghua Science and Technology*, vol. 26, no. 4, pp. 403–416, Aug. 2021.

[116] C. Zhang, Y. Zhu, C. Markos, S. Yu, and J. J. Q. Yu, "Toward crowdsourced transportation mode identification: A semisupervised federated learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 11 868–11 882, Jul. 2022.

[117] S. H. Hosseini, G. Gentile, L. M. B. Miristice, F. Viti, and S. Pourkhosro, "Deep neural networks for identifying modes of transport using GPS data," in *2024 IEEE International Conference on Environment and Electrical Engineering and 2024 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, Jun. 2024, pp. 1–5.

[118] A. Alazeb, D. Khan, and A. Jalal, "Intelligent transportation activity recognition using deep belief network," in *2024 4th Interdisciplinary Conference on Electrics and Computer (INTCEC)*, Jun. 2024, pp. 1–6.

[119] P. Sadeghian, X. Zhao, A. Golshan, and J. Håkansson, "A stepwise methodology for transport mode detection in GPS tracking data," en, *Travel Behaviour and Society*, vol. 26, pp. 159–167, Jan. 2022.

[120] Z. Xiao, Y. Wang, K. Fu, and F. Wu, "Identifying different transportation modes from trajectory data using tree-based ensemble classifiers," *ISPRS International Journal of Geo-Information*, vol. 6, no. 2, p. 57, Feb. 2017.

[121] X. Liang and G. Wang, "A convolutional neural network for transportation mode detection based on smartphone platform," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct. 2017, pp. 338–342.

[122] H. Gong, C. Chen, E. Bialostozky, and C. T. Lawson, "A GPS/GIS method for travel mode detection in new york city," *Computers, Environment and Urban Systems*, Special Issue: Geoinformatics 2010, vol. 36, no. 2, pp. 131–139, Mar. 1, 2012.

[123] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, "Big data small footprint: The design of a low-power classifier for detecting transportation modes," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, Aug. 2014.

[124] D. Shin, D. Aliaga, B. Tunçer, *et al.*, "Urban sensing: Using smartphones for transportation mode classification," *Computers, Environment and Urban Systems*, Special Issue on Volunteered Geographic Information, vol. 53, pp. 76–86, Sep. 1, 2015.

[125] V. Manzoni, D. Maniloff, K. Kloeckl, and C. Ratti, "Transportation mode identification and real-time CO2 emission estimation using smartphones," *SENSEable City Lab, Massachusetts Institute of Technology, nd*, 2010.

[126] S. Wang, C. Chen, and J. Ma, "Accelerometer based transportation mode recognition on mobile phones," in *2010 Asia-Pacific Conference on Wearable Computing Systems*, Apr. 2010, pp. 44–46.

[127] S. Lee, J. Lee, and K. Lee, "VehicleSense: A reliable sound-based transportation mode recognition system for smartphones," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2017, pp. 1–9.

[128] H. R. Eftekhari and M. Ghatee, "An inference engine for smartphones to preprocess data and detect stationary and transportation modes," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 313–327, Aug. 1, 2016.

[129] H. Xia, Y. Qiao, J. Jian, and Y. Chang, "Using smart phone sensors to detect transportation modes," *Sensors*, vol. 14, no. 11, pp. 20 843–20 865, Nov. 2014.

[130] L. Bedogni, M. Di Felice, and L. Bononi, "Context-aware android applications through transportation mode detection techniques," *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2523–2541, 2016.

[131] S.-H. Fang, H.-H. Liao, Y.-X. Fei, *et al.*, "Transportation modes classification using sensors on smartphones," *Sensors*, vol. 16, no. 8, p. 1324, Aug. 2016.

[132] S.-H. Fang, Y.-X. Fei, Z. Xu, and Y. Tsao, "Learning transportation modes from smartphone sensors based on deep neural network," *IEEE Sensors Journal*, vol. 17, no. 18, pp. 6111–6118, Sep. 2017.

[133] K. Can Cetin, H. Irem Turkmen, and M. Amac Guvensan, "N-gram based transport mode detection models for energy constrained devices," in *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug. 2021, pp. 1–5.

[134] C. Wang, H. Luo, F. Zhao, and Y. Qin, "Combining residual and LSTM recurrent networks for transportation mode detection using multimodal sensors integrated in smartphones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5473–5485, Sep. 2021.

[135] K. Sankaran, M. Zhu, X. F. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh, "Using mobile phone barometer for low-power transportation context detection," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '14, New York, NY, USA: Association for Computing Machinery, Nov. 3, 2014, pp. 191–205.

[136] G. Yanyun, Z. Fang, C. Shaomeng, and L. Haiyong, "A convolutional neural networks based transportation mode identification algorithm," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2017, pp. 1–7.

[137] Y. Qin, H. Luo, F. Zhao, C. Wang, J. Wang, and Y. Zhang, "Toward transportation mode recognition using deep convolutional and long short-term memory recurrent neural networks," *IEEE Access*, vol. 7, pp. 142 353–142 367, 2019.

[138] I. Drosouli, A. Voulodimos, and G. Miaoulis, "Transportation mode detection using machine learning techniques on mobile phone sensor data," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '20, New York, NY, USA: Association for Computing Machinery, Jun. 30, 2020, pp. 1–8.

[139] J. V. Jeyakumar, E. S. Lee, Z. Xia, S. S. Sandha, N. Tausik, and M. Srivastava, "Deep convolutional bidirectional LSTM based transportation mode recognition," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp '18, New York, NY, USA: Association for Computing Machinery, Oct. 8, 2018, pp. 1606–1615.

[140] S. Kumar, A. Damaraju, A. Kumar, S. Kumari, and C.-M. Chen, "LSTM network for transportation mode detection," *Journal of Internet Technology*, vol. 22, no. 4, pp. 891–902, Jul. 1, 2021.

[141] L. Wang and D. Roggen, "Sound-based transportation mode recognition with smartphones," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 930–934.

[142] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Determining transportation mode on mobile phones," in *2008 12th IEEE International Symposium on Wearable Computers*, Sep. 2008, pp. 25–28.

[143] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, 13:1–13:27, Mar. 2, 2010.

[144] R. C. Shah, C.-y. Wan, H. Lu, and L. Nachman, "Classifying the mode of transportation on mobile phones using GIS information," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14, New York, NY, USA: Association for Computing Machinery, Sep. 13, 2014, pp. 225–229.

[145] M. Kamalian, A. Taherkordi, A. H. Payberah, and P. Ferreira, "FogFLeeT: Fog-level federated transfer learning for adaptive transport mode detection," in *2024 IEEE International Conference on Cloud Engineering (IC2E)*, Sep. 2024, pp. 22–33.

[146] M. Kamalian, A. Taherkordi, and P. Ferreira, "TeLeGaIT: Transfer learning on fog for generalizable and real-time transport mode detection," in *2024 9th International Conference on Fog and Mobile Edge Computing (FMEC)*, Sep. 2024, pp. 196–203.

[147] H. I. Ashqar, M. H. Almannaa, M. Elhenawy, H. A. Rakha, and L. House, "Smartphone transportation mode recognition using a hierarchical machine learning classifier and pooled features from time and frequency domains," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 244–252, Jan. 2019.

[148] R. Saravanan and P. Sujatha, "A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Jun. 2018, pp. 945–949.

[149] P. Widhalm, P. Nitsche, and N. Brändie, "Transport mode detection with realistic Smartphone sensor data," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov. 2012, pp. 573–576.

[150] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022 022, Feb. 2019.

[151] H. Lu, M. Pinaroc, M. Lv, S. Sun, H. Han, and R. C. Shah, "Locomotion recognition using XGBoost and neural network ensemble," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ser. UbiComp/ISWC '19 Adjunct, New York, NY, USA: Association for Computing Machinery, Sep. 9, 2019, pp. 757–760.

[152] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2016, pp. 1310–1315.

[153] J. Iskanderov and M. A. Guvensan, "Breaking the limits of transportation mode detection: Applying deep learning approach with knowledge-based features," *IEEE Sensors Journal*, vol. 20, no. 21, pp. 12 871–12 884, Nov. 2020.

[154] H. Moreau, A. Vassilev, and L. Chen, "The Devil is in the Details: An Efficient Convolutional Neural Network for Transport Mode Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 202–12 212, Aug. 2022.

[155] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53 040–53 065, 2019.

[156] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56 855–56 866, 2020.

[157] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and IoT applications," in *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, Aug. 2017, pp. 1–8.

[158] M. Lin, W.-J. Hsu, and Z. Q. Lee, "Detecting modes of transport from unlabelled positioning sensor data," *Journal of Location Based Services*, vol. 7, no. 4, pp. 272–290, Dec. 1, 2013.

[159] T. S. Ajani, A. L. Imoize, and A. A. Atayero, "An overview of machine learning within embedded and mobile devices–optimizations and applications," *Sensors*, vol. 21, no. 13, p. 4412, Jan. 2021.

[160] T. Feng and H. J. Timmermans, "Comparison of advanced imputation algorithms for detection of transportation mode and activity episode using GPS data," *Transportation Planning and Technology*, vol. 39, no. 2, pp. 180–194, Feb. 17, 2016.

[161] F. Y. Osisanwo, J. E. T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: Classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, no. 3, pp. 128–138, 2017.

[162] R. Choudhary and H. K. Gianey, "Comprehensive review on supervised machine learning algorithms," in *2017 International Conference on Machine Learning and Data Science (MLDS)*, Dec. 2017, pp. 37–43.

[163] S. Pouyanfar, S. Sadiq, Y. Yan, *et al.*, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, 92:1–92:36, Sep. 18, 2018.

[164] P. Widhalm, M. Leodolter, and N. Brändle, "Top in the Lab, Flop in the Field? Evaluation of a Sensor-based Travel Activity Classifier with the SHL Dataset," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp '18, New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 1479–1487.

[165] O. Burkhard, H. Becker, R. Weibel, and K. W. Axhausen, "On the requirements on spatial accuracy and sampling rate for transport mode detection in view of a shift to passive signalling data," en, *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 99–117, May 2020.

[166] M. A. Shafique and E. Hato, "Travel mode detection with varying smartphone data collection frequencies," *Sensors*, vol. 16, no. 5, p. 716, May 2016.

[167] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, and D. Roggen, "Enabling reproducible research in sensor-based transportation mode recognition with the sussex-huawei dataset," *IEEE Access*, vol. 7, pp. 10 870–10 891, 2019.

[168] R. D. Das and S. Winter, "A fuzzy logic based transport mode detection framework in urban environment," *Journal of Intelligent Transportation Systems*, vol. 22, no. 6, pp. 478–489, Nov. 2018.

[169] H. Kaur, H. S. Pannu, and A. K. Malhi, "A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions," *ACM Computing Surveys*, vol. 52, no. 4, 79:1–79:36, Aug. 2019.

[170] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *JAIR*, vol. 16, pp. 321–357, Jun. 1, 2002. arXiv: `1106.1813 [cs]`.

[171] N. Chumerin and M. M. Van Hulle, "Comparison of two feature extraction methods based on maximization of mutual information," in *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, Sep. 2006, pp. 343–348.

[172] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*, Aug. 2014, pp. 372–378.

[173] L. Ladha and T. Deepa, "Feature selection methods and algorithms," *International journal on computer science and engineering*, vol. 3, no. 5, pp. 1787–1797, 2011.

[174] M. Z. Naser, "Do we need exotic models? engineering metrics to enable green machine learning from tackling accuracy-energy trade-offs," *Journal of Cleaner Production*, vol. 382, p. 135 334, Jan. 1, 2023.

[175] T. Parr, J. Hamrick, and J. D. Wilson, "Nonparametric feature impact and importance," *Information Sciences*, vol. 653, p. 119 563, Jan. 1, 2024.

[176] A. Janecek, W. Gansterer, M. Demel, and G. Ecker, "On the relationship between feature selection and classification accuracy," in *Proceedings of the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery at ECML/PKDD 2008*, PMLR, Sep. 11, 2008, pp. 90–105.

[177] O. Subasi, S. Ghosh, J. Manzano, B. Palmer, and A. Marquez, "Analysis and benchmarking of feature reduction for classification under computational constraints," *Machine Learning: Science and Technology*, vol. 5, no. 2, p. 020 501, Apr. 2024.

[178] D. Rengasamy, J. M. Mase, A. Kumar, *et al.*, "Feature importance in machine learning models: A fuzzy information fusion approach," *Neurocomputing*, vol. 511, pp. 163–174, Oct. 28, 2022.

[179] D. Rengasamy, B. C. Rothwell, and G. P. Figueredo, "Towards a more reliable interpretation of machine learning outputs for safety-critical systems using feature importance fusion," *Applied Sciences*, vol. 11, no. 24, p. 11 854, Jan. 2021.

[180] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 75–88, Dec. 1, 2019.

[181] J. Hestness, S. Narang, N. Ardalani, *et al.*, *Deep learning scaling is predictable, empirically*, Dec. 1, 2017. arXiv: 1712.00409. [Online]. Available: http://arxiv.org/abs/1712.00409.

[182] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li, *Geolife gps trajectory dataset - user guide*, Geolife GPS trajectories 1.1, Jul. 2011.

[183] H. Gjoreski, M. Ciliberto, L. Wang, *et al.*, "The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics With Mobile Devices," *IEEE Access*, vol. 6, pp. 42 592–42 604, 2018.

[184] M. Erdelić, T. Erdelić, and T. Carić, "Dataset for multimodal transport analytics of smartphone users - collecty," *Data in Brief*, vol. 50, p. 109 481, Oct. 1, 2023.

[185] A. Mathur, T. Zhang, S. Bhattacharya, *et al.*, "Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2018, pp. 200–211.

[186] A. R. Alharbi and M. A. Thornton, "Demographic group classification of smart device users," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2015, pp. 481–486.

[187] A. Nanchen, L. Meegahapola, W. Droz, and D. Gatica-Perez, "Keep sensors in check: Disentangling country-level generalization issues in mobile sensor-based models with diversity scores," in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '23, New York, NY, USA: Association for Computing Machinery, Aug. 29, 2023, pp. 217–228.

[188] G. M. Weiss and J. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[189] Z. Gong, P. Zhong, and W. Hu, "Diversity in machine learning," *IEEE Access*, vol. 7, pp. 64 323–64 350, 2019.

[190] S. Taherinavid, S. V. Moravvej, Y.-L. Chen, J. Yang, C. S. Ku, and P. L. Yee, "Automatic transportation mode classification using a deep reinforcement learning approach with smartphone sensors," *IEEE Access*, vol. 12, pp. 514–533, 2024.

[191] M. A. Guvensan, B. Dusun, B. Can, and H. I. Turkmen, "A Novel Segment-Based Approach for Improving Classification Performance of Transport Mode Detection," en, *Sensors*, vol. 18, no. 1, p. 87, Jan. 2018.

[192] Q. Tang and H. Cheng, "Feature pyramid biLSTM: Using smartphone sensors for transportation mode detection," *Transportation Research Interdisciplinary Perspectives*, vol. 26, p. 101 181, Jul. 1, 2024.

[193] T. Abdulazim, H. Abdelgawad, K. M. N. Habib, and B. Abdulhai, "Using smartphones and sensor technologies to automate collection of travel data," *Transportation Research Record*, vol. 2383, no. 1, pp. 44–52, Jan. 1, 2013.

[194] T.-T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, Sep. 1, 2015.

[195] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, Feb. 2016, pp. 78–83.

[196] A. Canziani, E. Culurciello, and A. Paszke, "Evaluation of neural network architectures for embedded systems," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[197] I. C. Cavalcante, R. I. Meneguette, R. H. Torres, *et al.*, "Federated System for Transport Mode Detection," en, *Energies*, vol. 15, no. 23, p. 9256, Jan. 2022.

[198] X. Ma, Y. Zhao, L. Zhang, H. Wang, and L. Peng, "When mobile terminals meet the cloud: Computation offloading as the bridge," *IEEE Network*, vol. 27, no. 5, pp. 28–33, Sep. 2013.

[199] X. Dai, I. Spasić, S. Chapman, and B. Meyer, "The state of the art in implementing machine learning for mobile apps: A survey," in *2020 SoutheastCon*, Mar. 2020, pp. 1–8.

[200] E. Fazeldehkordi and T.-M. Grønli, "A survey of security architectures for edge computing-based IoT," *IoT*, vol. 3, no. 3, pp. 332–365, Sep. 2022.

[201] B. Kar, W. Yahya, Y.-D. Lin, and A. Ali, "Offloading using traditional optimization and machine learning in federated cloud–edge–fog systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1199–1226, 2023.

[202] T. Stockx, B. Hecht, and J. Schöning, "SubwayPS: Towards smartphone positioning in underground public transportation systems," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '14, New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 93–102.

[203] S. Dhar, J. Guo, J. (Liu, S. Tripathi, U. Kurup, and M. Shah, "A survey of on-device machine learning: An algorithms and learning theory perspective," *ACM Transactions on Internet of Things*, vol. 2, no. 3, 15:1–15:49, Jul. 8, 2021.

[204] A. McIntosh, S. Hassan, and A. Hindle, "What can android mobile app developers do about the energy consumption of machine learning?" *Empirical Software Engineering*, vol. 24, no. 2, pp. 562–601, Apr. 1, 2019.

[205] X. Liu, "GLMLP-TRANS: A transportation mode detection model using lightweight sensors integrated in smartphones," *Computer Communications*, vol. 194, pp. 156–166, Oct. 1, 2022.

[206] D. Coskun, O. D. Incel, and A. Ozgovde, "Phone position/placement detection using accelerometer: Impact on activity recognition," in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Apr. 2015, pp. 1–6.

[207] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[208] Y. S. Park, L. Konge, and A. R. J. Artino, "The positivism paradigm of research," *Academic Medicine*, vol. 95, no. 5, p. 690, May 2020.

[209] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 1, 2007.

[210] S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decision Support Systems*, vol. 15, no. 4, pp. 251–266, Dec. 1, 1995.

[211] A. R. Hevner, "A three cycle view of design science research," *Scandinavian journal of information systems*, vol. 19, no. 2, p. 4, 2007.

[212] J. Venable, J. Pries-Heje, and R. Baskerville, "FEDS: A framework for evaluation in design science research," *European Journal of Information Systems*, vol. 25, no. 1, pp. 77–89, Jan. 1, 2016.

[213] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[214] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[215] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[216] L. Sthle and S. Wold, "Analysis of variance (ANOVA)," *Chemometrics and Intelligent Laboratory Systems*, vol. 6, no. 4, pp. 259–272, Nov. 1989.

[217] L. Breiman, *Classification and Regression Trees*. New York: Routledge, Oct. 2017.

[218] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Aug. 2020, pp. 729–735.

[219] X. Jiang and C. Xu, "Deep Learning and Machine Learning with Grid Search to Predict Later Occurrence of Breast Cancer Metastasis Using Clinical Data," *Journal of Clinical Medicine*, vol. 11, no. 19, p. 5772, Sep. 2022.

[220] N. Dong, M.-d. Zhai, L. Zhao, and C. H. Wu, "Cervical cell classification based on the CART feature selection algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1837–1849, Feb. 1, 2021.

[221] X. Li, W. Chen, Q. Zhang, and L. Wu, "Building Auto-Encoder Intrusion Detection System based on random forest feature selection," *Computers & Security*, vol. 95, p. 101 851, Aug. 2020.

[222] C. Chen, Q. Zhang, B. Yu, *et al.*, "Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier," *Computers in Biology and Medicine*, vol. 123, p. 103 899, Aug. 2020.

[223] H. Zhuang, X. Liu, H. Wang, *et al.*, "Diagnosis of early stage parkinson's disease on quantitative susceptibility mapping using complex network with one-way anova f-test feature selection," *Journal of Mechanics in Medicine and Biology*, vol. 21, no. 05, p. 2 140 026, Jun. 2021.

[224] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized Mutual Information Feature Selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[225] Y. Wang, J. Lin, M. Annavaram, *et al.*, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09, New York, NY, USA: Association for Computing Machinery, Jun. 22, 2009, pp. 179–192.

[226] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: Energy-efficient mobile sensing with senseless," in *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, ser. MobiHeld '09, New York, NY, USA: Association for Computing Machinery, Aug. 17, 2009, pp. 61–62.

Part II

Attachments

Chapter 7

Co-Author Declarations

7.1 Article I

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

***NB!** The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 1

Title of article: Survey of Automated Fare Collection Solutions in Public Transportation

Name of candidate: Anders Skretting

First author: Malte Bieler Shared first authorship: Anders Skretting Second author:

Philippe Büdinger Senior author: Tor-Morten Grønli

The independent contribution of the candidate: The candidate independently produced the chapters "Mobile applications and enabling technologies" and "IoT". The candidate also contributed significantly to the "Introduction", "Methodology", "Discussion" and "Outlook" chapters. The candidate also contributed with the overall planning and revisions.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/**
No:X

If yes, please name the co-author:____

Co-author: Malte Bieler

First author: X Shared first authorship:___Second author:___ Senior author:___ Other:___

The independent contribution of the co-author: Malte Bieler independently produced the chapter "Behavior mobility analysis" as well as contributing significantly to the "Introduction", "Methodology", "Discussion" and "Outlook" chapters. The co-author also contributed with the overall planning and revisions.

Must be signed by the candidate and co-author

Anders Skretting

Malte Bieler

Handwritten signature of candidate

Handwritten signature of co-author

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

NB! *The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 1

Title of article: Survey of Automated Fare Collection Solutions in Public Transportation

Name of candidate: Anders Skretting

First author: Malte Bieler Shared first authorship: Anders Skretting Second author:

Philippe Büdinger Senior author: Tor-Morten Grønli

The independent contribution of the candidate: The candidate independently produced the chapters "Mobile applications and enabling technologies" and "IoT". The candidate also contributed significantly to the "Introduction", "Methodology", "Discussion" and "Outlook" chapters. The candidate also contributed with the overall planning and revisions.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**:X

If yes, please name the co-author:____

Co-author: Philippe Büdinger

First author: Shared first authorship:___Second author: X Senior author:___ Other:___

The independent contribution of the co-author: Philippe Büdinger independently produced the chapter "Machine-learning use-cases with automated fare collection data", as well as contributing to the "Introduction", "Methodology", "Discussion" and "Outlook" chapters. The co-author also contributed with the overall planning and revisions.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Handwritten signature of co-author

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

NB! *The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 1

Title of article: Survey of Automated Fare Collection Solutions in Public Transportation

Name of candidate: Anders Skretting

First author: Malte Bieler Shared first authorship: Anders Skretting Second author:

Philippe Büdinger Senior author: Tor-Morten Grønli

The independent contribution of the candidate: The candidate independently produced the chapters "Mobile applications and enabling technologies" and "IoT". The candidate also contributed significantly to the "Introduction", "Methodology", "Discussion" and "Outlook" chapters. The candidate also contributed with the overall planning and revisions.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**:X

If yes, please name the co-author:____

Co-author: Tor-Morten Grønli

First author: Shared first authorship:___Second author:___ Senior author: X Other:___

The independent contribution of the co-author: Tor-Morten Grønli provided guidance and feedback during the work the article.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Handwritten signature of co-author

7.2 Article II

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

NB! *The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 2

Title of article: Neural Network for Public Transport Mode Inference on Mobile Devices

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli Other

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Tor-Morten Grønli

First author:___ Senior author: X Other:___

The independent contribution of the co-author: Tor-Morten Grønli provided guidance and feedback during all stages of the research, as well as during the production of the article.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Handwritten signature of co-author

7.3 Article III

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: “Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under.”

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

NB! *The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 3

Title of article: Enhanced Transport Mode Recognition on Mobile Devices

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli Other: Tim A. Majchrzak, Cristian Mateos, Matías Hirsch

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Tor-Morten Grønli

First author:___ Senior author: X Other:___

The independent contribution of the co-author: Tor-Morten Grønli provided guidance and feedback during all stages of the research, as well as during the production of the article.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Handwritten signature of co-author

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: “Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under.”

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

NB! *The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 3

Title of article: Enhanced Transport Mode Recognition on Mobile Devices

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli Other: Tim A. Majchrzak, Cristian Mateos, Matías Hirsch

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Tim A. Majchrzak

First author:___ Senior author:__ Other: X

The independent contribution of the co-author: Tim A. Majchrzak assisted in revising the article in terms of presentation.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Tim [illegible]

Handwritten signature of co-author

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

***NB!** The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 3

Title of article: Enhanced Transport Mode Recognition on Mobile Devices

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli Other: Tim A. Majchrzak, Cristian Mateos, Matías Hirsch

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Cristian Mateos

First author:___ Senior author:__ Other: X

The independent contribution of the co-author: Cristian Mateos assisted in revising the article in terms of presentation.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

CRISTIAN MATEOS DIAZ

Handwritten signature of co-author

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

***NB!** The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 3

Title of article: Enhanced Transport Mode Recognition on Mobile Devices

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli Other: Tim A. Majchrzak, Cristian Mateos, Matías Hirsch

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Matías Hirsch

First author:___ Senior author:__ Other: X

The independent contribution of the co-author: Matías Hirsch assisted in revising the article in terms of presentation.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Handwritten signature of co-author

Kristiania

7.4 Article IV

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

***NB!** The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 4

Title of article: Feature Importance from Novel Dataset for On-Device Transport Mode Detection

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli, Raghava Rao Mukkamala

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Tor-Morten Grønli

First author:___ Senior author: X

The independent contribution of the co-author: Tor-Morten Grønli provided guidance and feedback during all stages of the research, as well as during the production of the article.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Handwritten signature of co-author

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

NB! *The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 4

Title of article: Feature Importance from Novel Dataset for On-Device Transport Mode Detection

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli, Raghava Rao Mukkamala

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Raghava Rao Mukkamala

First author:___ Senior author: X

The independent contribution of the co-author: Raghava Rao Mukkamala provided guidance and feedback during all stages of the research, as well as during the production of the article.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

2025/01/30

Handwritten signature of co-author

7.5 Article V

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

***NB!** The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 5

Title of article: Implementation and Evaluation of Cross-Platform, Lightweight, On-Device Transport Mode Detection

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli, Raghava Rao Mukkamala

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Tor-Morten Grønli

First author:___ Senior author: X

The independent contribution of the co-author: Tor-Morten Grønli provided guidance and feedback during all stages of the research, as well as during the production of the article.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

Handwritten signature of co-author

Co-author declaration

Describing the independent research contribution of the candidate and each co-author

With reference to the Regulations for the degree of Philosophiae Doctor (PhD) at Kristiania University College § 11-3: "Med doktorgradsarbeid hvor det inngår bidrag fra flere, skal det følge en underskrevet erklæring som beskriver kandidatens innsats i hvert enkelt arbeid. Både kandidat og bidragsyter skal skrive under."

The co-author declaration must be filled in electronically and signed by the candidate and co-author. Only the five most important co-authors of an article have to sign the declaration. Each co-author must complete one co-author declaration. The candidate must sign each co-author declaration and must make sure that the declaration and signatures are on the same page.

NB! *The candidate must enclose the co-author declaration(s) with his/her application for thesis evaluation.*

Article no. : 5

Title of article: Implementation and Evaluation of Cross-Platform, Lightweight, On-Device Transport Mode Detection

Name of candidate: Anders Skretting

First author: Anders Skretting Senior author: Tor-Morten Grønli, Raghava Rao Mukkamala

The independent contribution of the candidate: The candidate conducted the research and produced the article independently.

To the best of your knowledge, has this article been part of a previously evaluated doctoral thesis?
Yes:___ **/ No**: X

If yes, please elaborate:____

____

Do you know if one of your co-authors is going to use this article in his/her doctoral thesis? Yes:___ **/ No**: X

If yes, please name the co-author:____

Co-author: Raghava Rao Mukkamala

First author:___ Senior author: X

The independent contribution of the co-author: Raghava Rao Mukkamala provided guidance and feedback during all stages of the research, as well as during the production of the article.

Must be signed by the candidate and co-author

Anders Skretting

Handwritten signature of candidate

2025/01/30

Handwritten signature of co-author

Part III

Articles

Kristiania

Chapter 8

Article I

Survey of Automated Fare Collection Solutions in Public Transportation

Publication status: Published in Journal of IEEE Transactions on Intelligent Transportation Systems, VOL. 23, NO. 9

Article I: Survey of Automated Fare Collection Solutions in Public Transportation –

DOI: 10.1109/TITS.2022.3161606

Kristiania

Chapter 9

Article II

Neural Network for Public Transport Mode Inference on Mobile Devices

Publication status: Published in Proceedings of 18th International Conference on Mobile Web and Intelligent Information Systems

Article II: Neural Network for Public Transport Mode Inference on Mobile Devices

DOI: 10.1007/978-3-031-14391-5_5

Kristiania

Chapter 10

Article III

Enhanced Transport Mode Recognition on Mobile Devices

Publication status: Published in Proceedings of the 57th Hawaii International Conference on System Sciences

Enhanced Transport Mode Recognition on Mobile Devices

Anders Skretting and Tor-Morten Grønli
School of Economics, Innovation, and Technology
Kristiania University College, Norway
{anders.skretting,tor-morten.gronli}@kristiania.no

Tim A. Majchrzak
University of Agder,
Kristiansand, Norway
timam@uia.no

Cristian Mateos and Matías Hirsch
ISISTAN (UNICEN-CONICET)
Tandil 7000, Argentina
{cristian.mateos,matias.hirsch}@isistan.unicen.edu.ar

Abstract

Establishing the context of users of mobile applications is essential to provide the users with relevant information and functionality associated with the user's location or situation. Context aware solutions adapt its behaviour to fit the situation the user is situated in by making certain information or functions available. They are applicable to most kinds of modern systems; public transportation systems are no exception. To achieve intelligent transportation, it is vital to determine contextual information related to travelers, such as which vehicle is currently used and where the travelers boarded or disembarked. This contributes towards more seamless public transport ticketing and provides public transport operators with enriched data. In this paper, we suggest an approach, using machine learning, to determine a traveler's mode of transport using mobile sensor data from the traveler's smartphone. The trained machine learning models can infer the mode of transport with high accuracy using off-the-shelf technology.

Keywords: machine learning, mobile, activity recognition, XGBoost, sensors, public transport, vehicle mode detection

1. Introduction

Lately, context awareness has become an important topic when designing most digital solutions, especially when it comes to mobile solutions. Since mobile users are inherently mobile, the environment where they are located is ever-changing and in that sense so is also the context of which they are situated in. Context awareness can be split into three different groups. Namely, computing context, user context and physical context (Liu et al., 2011). The different types of contexts can contribute to solving or improving different types of problems. For instance, by correctly establishing a computing context we can improve performance related to computing, such as waiting on available resources, for instance memory or bandwidth before starting a resource intensive process – e.g. a heavy IA task – in the mobile device or a nearby Edge server (Hirsch et al., 2021). The user context, on the other hand, is more related to the user's geographical location and what and whom are nearby. A system with Bluetooth capabilities could for instance establish, through scanning, that people are nearby and as such start a chat or a game with people nearby. Lastly, physical context is more related to physical variables around the user such as lighting, temperature and so forth. With context aware capabilities, mobile phones might increase or decrease its screen lighting based on the amount of natural lighting or change the content of the screen depending on whether the user holds the phone in portrait or landscape mode. In regards to physical context, by sensing the user's surroundings and the movement of a user's device, we can detect a user's type of physical activity or the type of location where the user is currently situated, such as within a house or a car. Solutions can in turn build on this to offer context specific functions. For instance, if a user is driving a car the smartphone might limit itself to only accept voice commands instead of touch input to increase safety. However, in order for systems to act and adapt based on the context we need to be able to quantify what distinguishes the different contexts, in addition to be able to infer context(s) changes.

Today, most mobile devices possess activity recognition capabilities. Activity recognition can be used in a wide variety of systems, such as fitness tracking, fall detection, health monitoring and so on (Lockhart et al., 2012). Activity recognition is, in essence, a built-in solution to establish a user's physical context, although at the moment most activity recognition solutions are fairly limited. Both Android and iOS smartphones have embedded functionality to infer whether a user is walking, running, bicycling and even if the user is situated in a vehicle. While these built-in activity recognition solutions are improving, more specific functionality, such as recognizing the type of vehicle, is still lacking. Most of the existing activity recognition solutions are based on machine learning (Shoaib et al.,

URI: https://hdl.handle.net/10125/107317
978-0-9981331-7-1

2015) and to classify a user's context we can take advantage of sensor data available through mobile phones (Oplenskedal et al., 2020; Shoaib et al., 2015; Skretting & Grønli, 2022). Context awareness enable and improve systems in a plethora of different aspects of our daily lives and mobility and public transport solutions could greatly benefit from enhanced activity recognition geared towards contextual changes during public transport journeys. Knowledge about the mode of transport can provide stakeholders and decision-makers in mobility and public transport companies with useful statistics which in turn can be taken advantage of when planning maintenance and new public transport routes. Furthermore, by automatically detecting whether a traveler is on board a public transport vehicle we can facilitate the implementation of more intelligent public transport solutions such as Be-In/Be-Out (BIBO) solutions where explicit interaction no longer are needed (Bitew et al., 2020; Oplenskedal et al., 2020; Servizi et al., 2022). In this paper, we propose a novel approach for transport mode recognition towards solving the issue of accurate in-vehicle presence detection by training machine learning models based on mobile sensor data. The models are able to infer the mode of transport a traveler is in with high accuracy, using only available sensor data from the traveler's smartphone.

Section 2 reviews efforts on activity recognition and solutions towards solving the challenge of accurate in-vehicle presence detection for intelligent public transport systems. Section 3 positions our work within the realm of public transport and provides details on data collection. Section 4 entails the analysis in terms of feature selection and balancing of the collected dataset, in addition to presenting the results of the machine learning models. In Section 5 we discuss our findings before concluding in Section 6.

2. Background

Neither context awareness, nor activity recognition are particularly new phenomenons and there have been numerous and significant research efforts towards both of these concepts. Bulling et al. (2014) defined an Activity Recognition Chain (ARC) which is a sequence of signal processing, pattern recognition and machine learning techniques that implements a specific activity recognition system behaviour. ARC entails steps such as preprocessing, segmentation feature extraction and classification. Although ARC bears strong resemblance to other pattern recognition systems, it also has a number of specific requirements and constraints. ARC has become one of the most prominent ways of handling activity recognition and many authors have employed this procedure in their work. One example is the work by Baldominos et al. (2019). In their paper they conducted a comparison of deep learning techniques for activity recognition using mobile devices. They tested out various classifiers such as extremely randomized trees, random forest, logistic regression, naive Bayes, K-nearest neighbours and multi-layer perceptron when attempting to classify various activities such as common postures, working activities and leisure activities to mention a few. They found that ensembles of decision trees, such as random forest, yielded the highest accuracy. Whether we are inferring physical activities such as running, or less active activities such as being a passenger, the same approach for data collection, preprocessing, training and classification can be employed.

Authors have employed similar techniques to solve the problem of accurate in-vehicle presence detection for BIBO solutions, however, most of these authors rely on on-board equipment. Wieczorek and Poniszewska-Marańda (2019) proposed to mount beacons inside public transport vehicles and to employ machine learning techniques to detect whether a passenger is actually situated inside the vehicle, within range of the beacon signal. Other authors have also attempted various approaches using beacons and Bluetooth scanners (Kostakos et al., 2010; Narzt et al., 2015, 2016), however Bluetooth seems to come short when high accuracy in-vehicle presence detection is required. Oplenskedal et al. (2020) instead proposed a framework where they took advantage of a reference device, in combination with a neural network to establish in-vehicle presence. In their approach, sensor data, such as acceleration, barometer, magnetic field and gyroscope are collected from the passengers' smartphones and is then compared with that of an on-board reference unit. The data from both devices is sent to a cloud solution which is running a neural network which in turn estimates the likelihood of a passenger being inside the given vehicle. The same authors have also improved this solution both in terms of in-vehicle presence detection accuracy and the amount of data transfer needed (Oplenskedal et al., 2022). The authors increased the accuracy through adapting the number of convolutional layers and filters and reduced the amount of data transfer by reducing the number of sensors needed to only require data from the barometer.

Skretting and Grønli (2022) proposed a mobile application consisting of a pre-trained multi-layer perceptron to infer the mode of transport. While most other attempts require equipment mounted inside the various transport vehicles, this approach would only utilize the travelers smartphone to classify the mode of transport. While this approach might be more cost efficient, the accuracy suffered slightly compared to the work by Oplenskedal et al. (2020). However, the authors did reach an accuracy of above 90 percent. In addition, since the inference was conducted on-device, no data transfer was required as opposed to the solution presented by Oplenskedal et al. (2020).

Although previous work close in on accurate solutions for in-vehicle presence detection of BIBO solutions, improvements can be made to accuracy, data transfer, architectural design and scalability.

3. Approach

Scalability and cost are important aspects if public transport operators are to implement an in-vehicle presence detection solution at large scale. Our approach requires no additional equipment and relies solely on the traveler's smartphone. By omitting the need for on-board equipment we can also reduce cost related to installation and maintenance. We have conducted a similar procedure to that of ARC (Bulling et al., 2014) in terms of data collection, preprocessing and classification. Even though transport mode recognition can be used in a variety of contexts, we focus on utilizing the solution towards solving the challenge of in-vehicle presence detection for intelligent public transport solutions.

3.1. Intelligent Public Transport

Traditionally, public transport solutions entailed a fairly manual process of obtaining and validating tickets. Different countries and locations worldwide have different implementations of public transport ticketing systems, however, most of them require some degree of manual operation in order for travelers to obtain a ticket. Whether the solution is digital, such as a mobile application, or physical, the traveler will always have to explicitly do a manual operation to obtain the ticket. Moreover, most of these solutions does not provide the public transport operators with sufficient data and statistics as to which routes are being used most frequently or even, which types of public transport vehicles are being utilized. Lately, researchers have been been looking into the concept of Be-In/Be-Out (BIBO) solutions where explicit interaction is no longer needed (Bieler et al., 2022; Bitew et al., 2020; Servizi et al., 2022). The idea is that travelers can embark or disembark any public transport vehicle and the system would automatically know which passenger traveled what distance with which public transport vehicle. As such, the system can automatically issue a ticket, or withdraw the required amount from the traveler's digital wallet or bank card. Employing a BIBO solution comes with many benefits. First and foremost, traveling using public transport will be seamless and effortless in terms of ticket purchase which can be a challenging task, especially when maneuvering in large, complicated networks with different zones and pricing schemes. Additionally, if the system knows the exact point of embarking and disembarking, new, more dynamic business models can be taken advantage of such as distance-based or duration-based pricing schemes. This can lead to fairer pricing, especially for people living close to bordering zones. Moreover, the information collected through this kind of solution can contribute to ease the decision-making process and lead to better decisions which may benefit the general population in terms of planning new roads, public transport routes and so on. BIBO solutions stand in contrasts to other public transport ticketing paradigms where passengers are required to explicitly check in or check out using either physical or digital tickets as opposed to BIBO which does not require any explicit interaction. BIBO solutions, however, needs to be able to accurately place a traveler within a given public transport vehicle and this is a non-trivial process.

3.2. Data Collection

Our approach takes advantage of machine learning techniques to establish a traveler's presence within a given mode of transport. The machine learning models were trained on a large amount of sensor data collected through actual travelers' smartphones. To establish a ground truth regarding which values could be expected of the different sensors while being situated within the different modes of transport we enlisted a total of 101 regular travelers in two different cities in Norway. These travelers had to install a custom mobile application which collected all available sensor data on the device when activated. Every traveler was instructed, in person, on how to operate the application so that we didn't risk any contamination of the dataset. In the custom mobile application, the traveler had to choose which mode of transportation they were currently using during the data collection, in addition to selecting where they would place the device. Once both transportation vehicle and device location were selected the traveler could activate the data collection. When activated, the application would check for available sensors and capture all sensor events on the device. Since transportation modes such as metro or train sometimes travel below ground the data was stored on the device so that it was not dependent on an active internet connection to collect the data. By storing the data on the devices we ensured that the dataset included sensor data from all areas where public transportation may venture, regardless of the level of

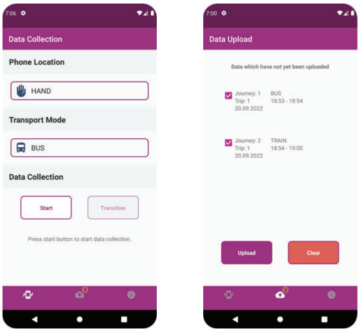


Figure 1. Custom Mobile Application

internet connectivity. After a journey, when a stable internet connection was available, the traveler would then upload the data.

Figure 1 presents the data collection view and the upload view of the custom mobile application. Both the phone placement selection and the transport mode selection are drop-down menus. For the phone placement the traveler could at any time choose between *hand*, *pocket* and *other*. Other was selected if the traveler put the device in a bag, backpack or similar. For the transport mode the traveler would choose either *bus*, *train*, *tram*, *metro*, *e-scooter*, *bicycle*, *boat*, *car*, *inside* or *outside*. All participants were instructed to activate the app while they were using any of the aforementioned transport modes. While most of these are self-explanatory both inside and outside might seem vague. The participants were instructed to use inside when they were located inside a building and *not* inside any vehicles. Outside was used when the participants were outside and *not* using any vehicles such as e-scooter or bicycle. We collected data labeled inside and outside to use as a control for the algorithm later on. The data collection resulted in a total of 576,839,057 sensor events of which 536,632,311 were from Android devices and 40,206,746 were from iOS devices. From Figure 2 we can see the distribution of sensor events for each transport mode, for both Android and iOS. As we can see, we collected significantly more data on board busses than any other mode of transport for both operating systems. For Android, the mode of transport where we collected the least amount of data was e-scooter; for iOS we did not collect any data for e-scooter (sensor data collected is presented in Table 1).

The sensor data was collected from a large variety of devices, representing both the major operating systems, iOS and Android, in addition to a diverse selection of models, both high-end and low-end. Figure 3 shows the distribution of different devices used to collect the dataset. By collecting sensor data from such a diverse set

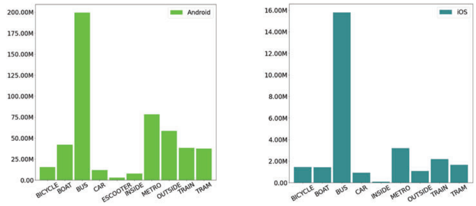


Figure 2. Sensor Event Distribution

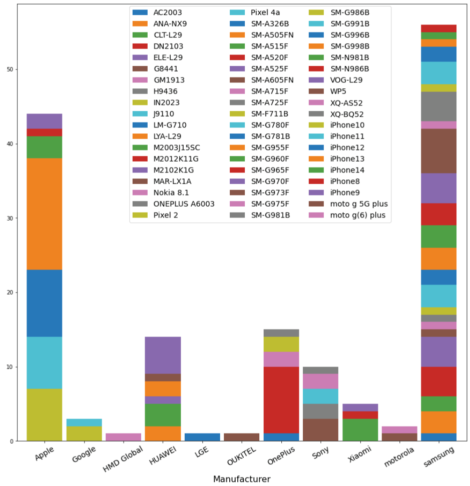


Figure 3. Distribution of Devices

of devices we can significantly limit the potential issues related to variations in software, hardware and sensor manufacturers, in addition to better represent the sensor data available in the general population.

The dataset consisted of a unique installation id, timestamp, mode of transport, type string and integer corresponding to the device sensor that generated the sensor event, the actual sensor data, the manufacturer and model, in addition to the physical location of the device. We also had a counter within the mobile application that incremented the number of journeys and trips taken which we appended to the sensor event. We defined a journey as going from the starting location to the final destination, while a trip is each individual mode of transport within a journey. Although we now had two quite large datasets, multiple steps of preprocessing were required to be able to effectively train machine learning models on it.

3.3. Data Preprocessing

Through the participants we had amassed two large datasets, one for iOS and one for Android. However, the raw data requires a series of processing techniques to be suitable for machine learning algorithms. From the collected data, in both datasets, we first removed empty data and fields containing `null` values for bad or missing measurements. Then we removed all trips with a duration less than 30 seconds or more than an hour. Trips shorter than thirty seconds were not likely to be an actual trip but rather an accidental data collection. Trips exceeding 1 hour, on the other hand, could be the result of a traveler forgetting to stop the data collection when leaving the public transport vehicle. For the Android data, we removed all device-specific sensors which were not part of the Android operating system. iOS devices does not possess any manufacturer specific sensors that is not part of the iOS operating system since Apple is both manufacturing the devices and providing the operating system. However, for both datasets we also removed several operating system specific sensors which were unlikely to yield any results based on previous research and intuition. For instance, most Android devices has a light sensor, however, light levels may vary based on the time of the day or where the device is placed (e.g. in a backpack) and as such, data relating to the light sensor were omitted. On iOS, on the other hand, we removed sensors such as proximity, screen brightness and so on. After removing sensors with a low likelihood of having a correlation with the mode of transport we were left with the following sensors for Android: accelerometer, magnetic field, orientation, gyroscope, barometer, gravity, linear acceleration, rotation vector and game rotation vector. Many of these sensors corresponds to sensors other authors has deemed useful in previous work (Mastalerz et al., 2020; Oplenskedal et al., 2020; Skretting & Grønli, 2022). For iOS we were left with accelerometer, barometer, motion quaternion, gyroscope, magnetometer, gravity and rotation rate. In addition, we also collected the amplitude of the surrounding noise through the device's microphone for both devices. Different modes of transportation could potentially have unique noise patterns which could help distinguish the traveler's mode of transport.

To investigate any particular patterns related to the movement and the environment of a particular public transport vehicle we analyzed the data over a period of time. We choose a sliding window of 10 seconds with a 5 seconds overlap based on previous research (Skretting & Grønli, 2022). This approach is similar to what other authors have done in their work (Baldominos et al., 2019; Bulling et al., 2014). The smaller the window size, the faster will the on-device inference be since the device will have to collect data over the same duration. However, with a too small window size it may be difficult to distinguish patterns related to the environment and patterns related to movement of the given transport mode. When aggregating the data over the chosen time frame we had to employ an aggregation function, however, the literature was not clear on what kind of aggregation function was most suitable for this kind of system so, with a window size of 10 seconds, we calculated the average, standard deviation, variance, max, min and the quantiles for each value, for both datasets. All the aforementioned steps resulted in a total of 260 different features for the Android dataset and 211 different features in the iOS dataset.

4. Result and Analysis

Even though we had reduced the dataset quite a bit after the aforementioned steps, it was still beneficial to reduce it even more. The less features required by the model, the less work is required by the mobile device to collect and process the data before running inferences. Moreover, by employing only the features that correlates the most with each mode of transport we can make sure that the performance of the model is as high as possible, in addition to reducing risk of overfitting. In our previous work we trained a multi-layer perceptron for transport mode classification (Skretting & Grønli, 2022) with good results, however for this work we alternatively took advantage of the XGBoost algorithm to see if this would lead to better performance for our our dataset. XGBoost has previously proven to yield good results when classifying mode of transport (Gertz et al., 2020; Lu et al., 2019).

4.1. Feature Selection

Now that we had selected a handful of sensors based on initial selection from previous research we wanted to examine whether removing any more would greatly impact the accuracy of the XGBoost model. So for both Android and iOS we trained a model using all available features and then trained new models where we removed one sensor at the time. Table 1 presents the results from the sensor removal experimentation.

From Table 1 we can clearly see which sensors cause a drop in the accuracy when removed. With a baseline of 96.86 % accuracy for Android we see that the two features that causes the accuracy to rise when removed is the game rotation vector and orientation. However, with a difference of only 0.06% for the game rotation vector and only 0.04% for the orientation, this could be a result of the stochastic nature of the algorithm. We can also see that removing the linear acceleration sensor causes the smallest drop in accuracy, with a drop of only 0.02%. Linear acceleration is a composite sensor in Android and is derived from the accelerometer, and since

Table 1. Sensor Removal Experimentation

Without sensor	Android model accuracy %	iOS model accuracy %
Accelerometer	96.68	96.99
Magnetometer	95.26	87.51
Orientation	96.90	N/A
Gyroscope	95.66	97.05
Barometer	95.99	95.87
Gravity	96.82	97.08
Linear acceleration	96.84	N/A
Rotation vector	96.76	N/A
Game rotation vector	96.92	N/A
Motion quaternion	N/A	96.73
Rotation rate	N/A	97.11
Audio	96.49	96.91
All sensors	96.86	97.08

the accelerometer is already a feature in our model, the need for linear acceleration might not be present. Audio, rotation vector, acceleration, linear acceleration, and gravity all causes a drop of less than 0.4% when removed. The magnetometer, gyroscope and barometer, on the other hand, causes the largest drop in accuracy when removed which attests to the findings in (Oplenskedal et al., 2022) where the authors found that the barometer was enough to infer in-vehicle presence detection. The drop in accuracy is very small for most sensors which is an indication that we might not need all of the sensors to reach high accuracy.

For the iOS dataset, on the other hand, we only achieved a higher accuracy by removing the rotation rate. However, since this was an increase of only 0.03%, this as well could just be a result of the stochastic nature of the algorithm. On iOS the importance of the individual sensors seems to differ slightly from the importance of the Android sensors. Removing audio, gyroscope or accelerometer caused negligible accuracy drop, while removing gravity caused no effect. However, removing the magnetometer caused the largest drop in accuracy, with a drop of almost 10% which indicates that the magnetometer might be more important on iOS devices when inferring mode of transport. Similar to Android, the barometer caused a notable drop in accuracy when removed. Note that we only removed one sensor at a time, trained a new model, added it again, before removing another. As such, the accuracy might be very different if we had chosen to remove multiple sensors at the same time. Since our goal for this work was to reach as high accuracy as possible and the accuracy increase of dropping sensors were insignificant we selected all of the sensors for our final models, both for Android and iOS.

In addition to experimenting with the different

Table 2. Aggregation Function Removal Experimentation

Without aggregation function	Android model accuracy %	iOS model accuracy %
Min	96.60	97.02
Max	96.80	97.01
1st quantile	96.75	96.98
2nd quantile	96.74	97.12
3rd quantile	96.71	96.97
Average	96.73	97.12
Range	96.91	97.18
Variance	96.89	97.01
Standard deviation	96.76	97.19
All calculations	96.79	97.08

Table 3. Aggregation Function Experimentation

Aggregation function	Android model accuracy %	iOS model accuracy %
Min	94.58	94.76
Max	94.97	95.23
1st quantile	92.71	95.50
2nd quantile	92.98	95.25
3rd quantile	93.26	95.27
Average	92.34	94.78
Range	92.06	84.73
Variance	87.03	82.89
Standard deviation	86.95	82.83

sensors, we conducted the same approach when evaluating the aggregation functions. We built a model with all of the aggregation functions and then removed one function, re-trained the model, added it again before removing another aggregation function. For Android we see that removing the range, max, variance or average causes an increase the accuracy, while removing the other aggregation functions caused a negligible drop in the accuracy. Surprisingly, removing the min value causes the largest drop in accuracy, however, since the variations are so minuscule it might just be the result of the stochastic nature of the algorithm. Similar to that of Android, for iOS the variation in accuracy is insignificant for all of the aggregation functions with the highest difference of 0.11% by removing the third quantile. Since we were not satisfied with the result of the aggregation function removal procedure (Table 2), we instead trained XGBoost models using only one of the aggregation functions at a time (Table 3).

By training models using only one aggregation function at a time we achieved greater variation in the results. For Android we see that the min and the max values yields the highest accuracy. Min and max is just

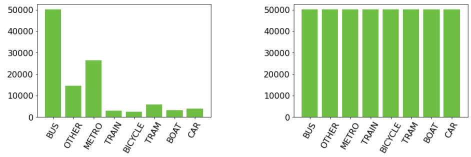


Figure 4. Data Distribution Before and After Applying SMOTE Android

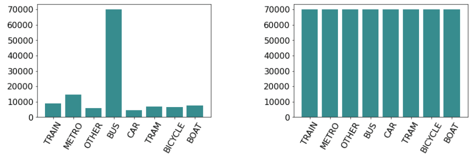


Figure 5. Data Distribution Before and After Applying SMOTE iOS

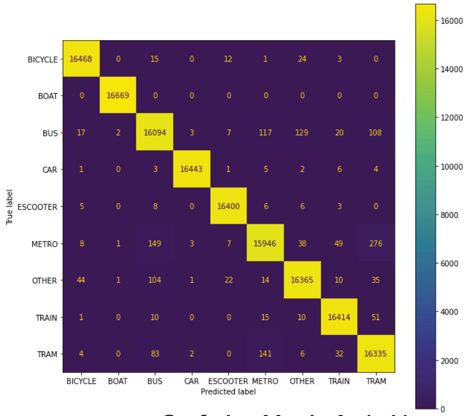


Figure 6. Confusion Matrix Android

the lowest and highest value observed within the 10 second window and is not really an aggregation function in and of itself. It is therefore quite surprising that using these functions yields a higher accuracy than functions such as standard deviation, average, variance and range. Both standard deviation and variance yielded the poorest accuracy. The results for iOS are quite similar where the standard deviation, variance and range performs least well. The rest of the functions perform similarly; however, for iOS, using the first quantile function for aggregation yields the highest accuracy, as opposed to the max value of Android. Nevertheless, since the variations in accuracy are tiny, this might be a result of the stochastic nature of the algorithm. From the aggregation function experimentation it is apparent that we achieve the highest accuracy using a combination of aggregation functions. Even though many of the aggregation functions seems to lower the accuracy of the models, we achieved the highest accuracy when combining them. Thus, we decided to train our final models by combining all the functions.

4.2. Imbalance

Figure 2 shows the distribution of sensor events within the different modes of transport. The dataset is fairly skewed and imbalanced. With imbalanced datasets there might be too few samples of the smaller classes for a model to learn the decision boundary effectively. We therefore had to address this issue before training our final models and as such took advantage of the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002) to balance our dataset. Figures 4 and 5 shows the distribution of data for the different classes before and after applying SMOTE for both Android and iOS.

For Android over 45% of our sensor events were captured on board busses, roughly 24% on metro, while events captured on board boats, cars, e-scooters, bicycle and trains resulted in less than 15% combined. The rest of the dataset consisted of sensor events captured outside and inside, which we grouped together as *other*, which in turn constituted roughly 24%. On iOS almost 56% of the data was captured on board busses, roughly 12% on metro, around 5% other, while the remaining modes accounted for the remaining 27%. When employing SMOTE, we synthesize data for the minority classes, resulting in every class having the same amount of values as the majority class, as seen in the figures 4 and 5.

4.3. Android

As our goal was to reach as high accuracy as possible without the use of any equipment, mounted in vehicles or otherwise, we used all the features from the feature evaluation and all the aggregation functions since this combination yielded the highest accuracy, not counting negligible variation. We then applied SMOTE to balance the dataset and reached an accuracy of 98.91% for the Android model. The confusion matrix for the Android model can be seen in Figure 6.

The matrix presents the number of correct predictions for each class, in this case, mode of transportation. From the confusion matrix we can see that the model is successful in classifying all modes of transport with high accuracy, with only a handful of wrong classifications for each class. Table 4 shows the report and we can see the precision. The classes with best performance was bicycle, boat, car and e-scooter, while bus, metro and tram performed the least well. From Table 4 we see that the final Android model has a precision, recall and f1-score of above 97% for all classes.

Table 4. **Classification Report Android**

	Precision	Recall	F1-score	Support
BICYCLE	0.99	1.00	1.00	16523
BOAT	1.00	1.00	1.00	16669
BUS	0.98	0.99	0.98	16497
CAR	1.00	1.00	1.00	16465
ESCOOTER	1.00	1.00	1.00	16428
METRO	0.98	0.97	0.97	16477
OTHER	0.99	0.99	0.99	16596
TRAIN	0.99	0.99	0.99	16501
TRAM	0.97	0.98	0.98	16603
accuracy			0.99	148759
micro avg	0.99	0.99	0.99	148759
weight. avg	0.99	0.99	0.99	148759

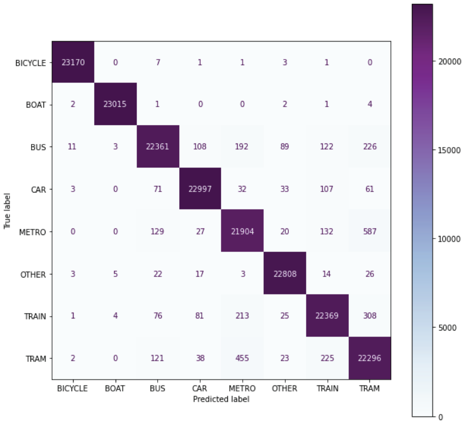


Figure 7. Confusion Matrix iOS

Table 5. **Classification Report iOS**

	Precision	Recall	F1-score	Support
BICYCLE	1.00	1.00	1.00	23183
BOAT	1.00	1.00	1.00	23025
BUS	0.98	0.97	0.97	23112
CAR	0.99	0.99	0.99	23304
METRO	0.96	0.96	0.96	22799
OTHER	0.99	1.00	0.99	22898
TRAIN	0.97	0.97	0.97	23077
TRAM	0.95	0.96	0.96	23160
accuracy			0.98	184558
micro avg	0.98	0.98	0.98	184558
weight. avg	0.98	0.98	0.98	184558

4.4. iOS

For the iOS model we employed the same approach and employed all sensors and aggregation functions. The final iOS model reached an accuracy of 98.03%. Note that for iOS we did not have any data on e-scooters so the e-scooter class was omitted for this model. The confusion matrix for the iOS model can be seen in Figure 7. The confusion matrix for iOS shows that the model is able to classify the different modes of transport with high accuracy, similar to that of the Android model.

Table 5 presents the classification report for the iOS model. The report is similar to that of the Android model, however it performs slightly worse when classifying most of the classes, except for bicycle, boat and other. The iOS model performs almost 1% less accurate than the Android model. However, for the iOS model we had significantly less data and different sensors.

5. Discussion

We proposed two machine learning models based on the XGBoost algorithm for transport mode recognition. Our models are able to successfully classify whether a traveler is situated in a bus, train, tram, metro, car, boat or bicycle, in addition to classifying when a traveler is *not* situated within any of these modes. Accurate transport mode recognition can be used as a component to solve the issue of in-vehicle presence detection for Be-In/Be-Out solutions. In-vehicle presence detection is a major challenge and a requirement to implement BIBO solutions. Without presence detection, there is no way of knowing whether the traveler is in fact using the public transport vehicle as opposed to walking next to it or driving behind it. Unless the system is able to determine, with high accuracy, the onboard presence of a traveler it can not automatically issue tickets and charge the travelers in a seamless fashion.

We achieved an accuracy of 98.91% and 98.03% for Android and iOS, respectively, which outperforms previous work (Oplenskedal et al., 2022). Moreover, our approach require no on-board equipment, and only requires sensor data input to infer the mode of transport. In dialog with various public transport operators in Norway we found that it was desirable to avoid installing any form of equipment on-board public transport vehicles to increase scalability while also reducing cost. Some previous work (Oplenskedal et al., 2022; Oplenskedal et al., 2020) employs an intricate distributed architecture where the traveler's device need to communicate with the onboard device during the trip. This results in an increased amount of data transmissions, which in turn may lead to an increase of power consumption on the traveler's device. Since our approach only consists of a single model, it could be deployed in a variety of different architectures. For instance, the models could be deployed in a cloud solution exposing an API routing mobile sensor data to the model and then returns the result of the inference. The travelers' could then collect the required data over a 10 second period, send it to the cloud solution to determine the mode of transport. Public transport operators could in turn use this to issue tickets, gather travel statistics and so on. Due to the flexibility of our approach we could also, instead, place the models within the public transport operators' mobile ticketing applications. By taking this approach the device would not need to transmit any data other than the result of the inference. Having the model placed on device would also improve privacy since no sensor data would have to leave the device other than the result of the inference. If the sensor data were to be intercepted malicious actors could potentially take advantage of the mobile sensor data to discern whether a traveler is home or not before conducting a break-in, or to derive a more specific location of the traveler.

While our approach has many benefits it is worth discussing potential issues as well. For instance, the proposed models of this approach require microphone access. In both Android and iOS this is an explicit permission the traveler would have to give in order for the system to be able to access it. It might be hard to convince a traveler of giving a public transport ticketing app access to their microphone, even though, in our solution, we do not capture full audio but only the amplitude or the noise level of the traveler's surroundings. It could therefore be beneficial to remove the dependency on the audio data if this solution were to be implemented on a larger scale. Nevertheless, our feature selection shows that without using the audio data we still achieve an accuracy of over 96% for both models, before applying SMOTE. The accuracy without audio was less than 0.3% lower than when including it for both models and we can therefore assume that the accuracy drop of omitting the audio values and applying SMOTE would result in an accuracy very close to what we achieved with audio.

Another aspect worth discussing is the reliability of the sensor data we collected through the 101 participants since this is the foundation for our models and serves as the ground truth. Although it is impossible to know for certain whether the participants labeled data with wrong transport mode, we went to great lengths to ensure that the data collected was as accurate as possible. Every participant had to attend a 2-hour in-person training session. We explained the importance of the data collection, carefully explained how to operate the mobile application and even did multiple trips afterwards with all the participants so that they could test out the solution under supervision. Every participant was awarded with a one month free public transport travel pass midway during the data collection period of four weeks. We decided to grant them the travel pass midway so that participants could not just show up, get their pass and then refuse to attend. During the in-person training session we also tried to assess the individual participant's motivation for doing this and participants who showed a lack of interest or motivation were rejected and denied access to the study. However, regardless of good intentions and motivation the participants could, of course, mistakenly press the wrong mode of transport or forget to stop the collection. We tried to mitigate this in three steps: First we stored all data on-device. Thereby, we removed the need for a stable internet connection and, as such, ensured that no data was lost in tunnels or similar. Furthermore, since the data was stored on the device we could aggregate and present the different trips for the participants in the mobile application so that they could verify that they had actually used the given mode of transport in the displayed time frame. In addition, during preprocessing of the dataset we removed all trips shorter than 30 seconds and longer than an hour to further mitigate the risk of participants forgetting to stop the data collection. Lastly, we carried out support during the whole period so that participants could report isuses, or if they had mistakenly uploaded wrong data that we should delete. All in all, we are very confident in the dataset and the ground truth it provides for our models.

Although we are confident in the dataset we collected and the large amount of sensor values we used for training the models, there is always a risk of overfitting, which is when a model does not generalize well from observed data to unseen data (Ying, 2019). When a machine learning model is trained with too much data it can learn the detail and noise in the training data so that it impacts the performance of the model negatively when presented with new data. That is to say, the model learns to classify based on the noise and arbitrary variation in the training data which does not necessarily apply to new data which in turn leaves the model with poor ability to generalize. However, we trained our models with only two thirds of the data and kept a holdout set of one third for testing and verification. As such, the accuracy presented for the different models attests to our models ability to generalize and is a strong indication that an overfitting problem is not present. Nevertheless, it could be that our models, have picked up and learned concepts based on the noise and seemingly random fluctuations in the dataset. However, we would then argue that the noise and fluctuations are not, in fact, random after all but rather a generalizable aspect of mobile sensor data.

6. Conclusion and Future Work

In this paper we presented two machine learning models, one for Android and one for iOS, based on the XGBoost algorithm for transport mode recognition. We achieved an accuracy of 98.91% and 98.03% for Android and iOS, respectively, using off-the-shelf technology. Our models are able to successfully classify transport modes such as bus, train, tram, metro, car, boat or bicycle, in addition to classifying when a traveler is *not* situated within any of these modes. For Android, the model is also able to classify whether a traveler is on an e-scooter. Our approach requires no on-board equipment and can be deployed either on-device or in a cloud solution. By omitting the need for any equipment we potentially reduce cost related to both installation and maintenance, not to mention the equipment in itself. When deploying the solution on device, no data transmission is needed and as such the privacy of the traveler is ensured.

In the future we would like to explore the dataset further to see if we can find other interesting correlations. Future work should entail testing out different machine learning algorithms and different balancing mechanisms. In addition, work should go into evaluating and finding optimal aggregation functions for these kinds of problems. Moreover, future work should go into implementing and evaluating the power consumption of this sort of algorithm running on a device. This for example can be done using a hard-soft approach to batch benchmarking smartphone (Yannibelli et al., 2023).

References

Baldominos, A., Cervantes, A., Saez, Y., & Isasi, P. (2019). A Comparison of Machine Learning and Deep Learning Techniques for Activity Recognition using Mobile Devices. *Sensors*, *19*(3), 521. https://doi.org/10.3390/s19030521

Bieler, M., Skretting, A., Büdinger, P., & Grønli, T.-M. (2022). Survey of Automated Fare Collection Solutions in Public Transportation. *IEEE Trans. Intell. Transp. Syst.*, *23*(9), 14248–14266. https://doi.org/10.1109/TITS.2022.3161606

Bitew, M. A., Muhammad, A., Fandiantoro, D. H., Boedinoegroho, H., & Kurniawan, A. (2020). E-Payment For Public Transportation Using BIBO Method Based On Bluetooth Low Energy Beacon. *Proc. CENIM '20*, 199–204. https://doi.org/10.1109/CENIM51130.2020.9297901

Bulling, A., Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, *46*(3), 33:1–33:33. https://doi.org/10.1145/2499621

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *JAIR*, *16*, 321–357. https://doi.org/10.1613/jair.953

Gertz, M., Große-Butenuth, K., Junge, W., Maassen-Francke, B., Renner, C., Sparenberg, H., & Krieter, J. (2020). Using the XGBoost algorithm to classify neck and leg activity sensor data using on-farm health recordings for locomotor-associated diseases. *Comput Electron Agric*, *173*, 105404. https://doi.org/10.1016/j.compag.2020.105404

Hirsch, M., Mateos, C., Zunino, A., Majchrzak, T. A., Grønli, T.-M., & Kaindl, H. (2021). A simulation-based performance evaluation of heuristics for dew computing. *Proc. HICSS-54*.

Kostakos, V., Camacho, T., & Mantero, C. (2010). Wireless detection of end-to-end passenger trips on public transport buses. *13th IEEE ITSC*, 1795–1800. https://doi.org/10.1109/ITSC.2010.5625062

Liu, W., Li, X., & Huang, D. (2011). A survey on context awareness. *CSSS*, 144–147. https://doi.org/10.1109/CSSS.2011.5972040

Lockhart, J. W., Pulickal, T., & Weiss, G. M. (2012). Applications of mobile activity recognition. *Proc. UbiComp '12*, 1054–1058. https://doi.org/10.1145/2370216.2370441

Lu, H., Pinaroc, M., Lv, M., Sun, S., Han, H., & Shah, R. C. (2019). Locomotion recognition using XGBoost and neural network ensemble. *Proc. UbiComp/ISWC '19 Adjunct*, 757–760. https://doi.org/10.1145/3341162.3344870

Mastalerz, M. W., Malinowski, A., Kwiatkowski, S., Śniegula, A., & Wieczorek, B. (2020). Passenger BIBO detection with IoT support and machine learning techniques for intelligent transport systems. *Procedia Computer Science*, *176*, 3780–3793. https://doi.org/10.1016/j.procs.2020.09.009

Narzt, W., Mayerhofer, S., Weichselbaum, O., Haselböck, S., & Höfler, N. (2015). Be-In/Be-Out with Bluetooth Low Energy: Implicit Ticketing for Public Transportation Systems. *18th IEEE ITSC*, 1551–1556. https://doi.org/10.1109/ITSC.2015.253

Narzt, W., Mayerhofer, S., Weichselbaum, O., Haselböck, S., & Höfler, N. (2016). Bluetooth Low Energy as enabling technology for Be-In/Be-Out systems. *13th IEEE CCNC*, 423–428. https://doi.org/10.1109/CCNC.2016.7444817

Oplenskedal, M., Herrmann, P., & Taherkordi, A. (2022). DeepMatch2: A comprehensive deep learning-based approach for in-vehicle presence detection. *Information Systems*, *108*, 101927. https://doi.org/10.1016/j.is.2021.101927

Oplenskedal, M., Taherkordi, A., & Herrmann, P. (2020). DeepMatch: Deep matching for in-vehicle presence detection in transportation. *Proc. DEBS '20*, 97–108. https://doi.org/10.1145/3401025.3401741

Servizi, V., Persson, D. R., Pereira, F. C., Villadsen, H., Bækgaard, P., Rich, J., & Nielsen, O. A. (2022). *Large Scale Passenger Detection with Smartphone/Bus Implicit Interaction and Multisensory Unsupervised Cause-effect Learning* [Preprint: http://arxiv.org/abs/2202.11962].

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., & Havinga, P. J. M. (2015). A Survey of Online Activity Recognition Using Mobile Phones. *Sensors*, *15*(1), 2059–2085. https://doi.org/10.3390/s150102059

Skretting, A., & Grønli, T.-M. (2022). Neural Network for Public Transport Mode Inference on Mobile Devices. *Mobile Web and Intelligent Information Systems*, 65–78. https://doi.org/10.1007/978-3-031-14391-5_5

Wieczorek, B., & Poniszewska-Marańda, A. (2019). Be in/Be out model for intelligent transport in SmartCity approach. *Proc. 17th MoMM2019*, 226–230. https://doi.org/10.1145/3365921.3365945

Yannibelli, V., Hirsch, M., Toloza, J., Majchrzak, T. A., Zunino, A., & Mateos, C. (2023). Speeding up smartphone-based dew computing in-vivo experiments setup via an evolutionary algorithm. *Sensors*, *23*(3).

Ying, X. (2019). An Overview of Overfitting and its Solutions. *J. Phys.: Conf. Ser.*, *1168*, 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

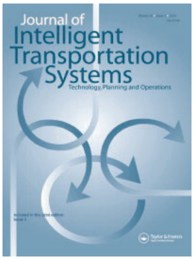
Kristiania

Chapter 11

Article IV

Feature Importance from Novel Dataset for On-Device Transport Mode Detection

Publication status: Published in Journal of Taylor & Francis Intelligent Transportation Systems



Journal of Intelligent Transportation Systems
Technology, Planning, and Operations

ISSN: 1547-2450 (Print) 1547-2442 (Online) Journal homepage: www.tandfonline.com/journals/gits20

Feature importance from novel dataset for on-device transport mode detection

Anders Skretting, Tor-Morten Grønli & Raghava Rao Mukkamala

JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS
https://doi.org/10.1080/15472450.2025.2515567

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Feature importance from novel dataset for on-device transport mode detection

Anders Skretting[a], Tor-Morten Grønli[a], and Raghava Rao Mukkamala[b]

[a]School of Economics, Innovation, and Technology, Kristiania University of Applied Sciences, Oslo, Norway; [b]Centre for Business Data Analytics, Copenhagen Business School, Frederiksberg, Denmark

ABSTRACT

Urbanization has increased reliance on intelligent transportation systems, with transport mode detection playing a pivotal role. Transport mode detection leverages sensor data from mobile devices to classify transportation modes, enabling applications such as optimized route planning and autonomous ticketing. However, existing datasets for developing accurate transport mode detection solutions often lack diversity in devices, participants, and transportation modes, limiting their generalizability. Feature selection in transport mode detection research also remains inconsistent, relying heavily on domain-specific experimentation. This study introduces a novel dataset addressing these limitations by including data from 101 participants across 57 unique device models, including both Android and iOS, and covering 10 transportation modes. Additionally, we present an ensemble-based framework for feature evaluation and reduction, which systematically ranks features in a generic and transferable manner. Evaluation shows that models trained using features ranked by the proposed framework achieve up to a 75% reduction in feature size while maintaining competitive accuracy, enabling efficient, on-device transport mode detection solutions. The framework also identifies the most impactful sensors and aggregation functions, offering insights transferable across diverse algorithms and applications.

ARTICLE HISTORY
Received 20 December 2023
Revised 27 May 2025
Accepted 1 June 2025

KEYWORDS
activity recognition; Android; dataset; iOS; machine learning; mobile; sensors; transport mode detection

Introduction

Urbanization is reshaping the planet, marking one of the most significant shifts in human settlement patterns in history. The rapid migration of people to cities, driven by the promise of better opportunities and economic growth, has transformed urban areas into the defining hubs of modern life, with profound social, economic, and environmental implications (Zhang, 2016). With the rapid expansion of urban areas and the increasing concentration of populations in cities (Angel et al., 2011), there is a growing need for efficient, sustainable, and inclusive transportation systems. Advances in sensor and network technology, combined with the widespread adoption of smartphones, have enabled the collection of vast amounts of data on vehicles and traveler behavior (Prelipcean et al., 2017; Zhang et al., 2011). These data provides a foundation for real-time insights into traffic patterns, delays, and mobility trends, allowing intelligent transportation solutions to enhance the efficiency and user experience of public transportation. Among these innovations, transport mode detection plays a pivotal role, offering the ability to automatically classify travelers' modes of transportation and support applications such as autonomous ticketing and optimized route planning (Mangiaracina et al., 2017; Oplenskedal et al., 2021; Zhang et al., 2011). Transport mode detection revolves around discerning which type of vehicle a user is situated in, using different combinations of contextual information such as location data, inertial or ambient sensor values collected from smartphones, smartwatches, or similar devices. This data is then used to train machine learning algorithms which are able to detect specific patterns that characterizes different modes of transportation. Different vehicles exhibit unique speed and acceleration patterns, which can be effectively captured by accelerometers in smartphones to distinguish between various modes of transportation. Additionally, barometric pressure sensors (barometers) can detect subtle height variations along a road, differences imperceptible to the

CONTACT Anders Skretting anders.skretting@kristiania.no School of Economics, Innovation, and Technology, Kristiania University of Applied Sciences, Oslo, Norway

naked eye, making them particularly useful for generating data indicative of specific transportation modes (Sankaran et al., 2014). Similarly, magnetometers measure the magnetic field surrounding the device and these readings are influenced by the structural composition of the vehicle, creating distinct magnetic signatures that can be leveraged to identify a person's mode of transportation (Chen et al., 2017). Modern smartphones are uniquely positioned to provide the data needed for transport mode detection solutions, due to their integrated sensors and widespread adoption (Cianciulli et al., 2017; Wahlström et al., 2017). Today, mobile devices represent the largest source of data on human transportation patterns, supporting advanced intelligent transportation solutions (Zannat & Choudhury, 2019) and this data forms the foundation for a large spectrum of innovative solutions in intelligent transportation systems.

Transport mode detection rely heavily on machine learning techniques and algorithms, which in turn depend on large, diverse, and representative datasets to accurately classify a user's vehicle or activity. However, collecting such datasets is often a costly, time-consuming, and complex process, as the data must be gathered from real travelers using a wide range of devices to ensure representativeness. An additional challenge is ensuring the collected data is accurately labeled to match the correct transport mode. This labeling process is typically performed by the data collector and must occur simultaneously with data collection to minimize uncertainty or errors in labeling. Currently, very few publicly available datasets for transport mode detection exist, and those that do are often limited in scope. Many suffer from homogeneity in terms of the devices used, the number of participants involved, or the range of sensors and transportation modes included. Notably, to the best of our knowledge, no publicly available datasets include data from iOS devices, despite these devices representing a significant share of the smartphone market (Grossi, 2019). The absence of certain transportation modes is another critical limitation. For example, if a dataset includes only buses and trains as labeled classes, the algorithm can only distinguish between these modes, making it impossible to classify other types of vehicles, such as trams or cars. Device diversity is also crucial, as sensor data can vary significantly between manufacturers and operating systems (Lane et al., 2010; Mathur et al., 2018) and training models on a dataset limited to specific devices may result in solutions that fail to generalize across a broader range of hardware. Therefore, data collection must include a diverse pool of devices, allowing algorithms to learn the differences in sensor data as they correspond to various manufacturers and systems. Furthermore, the way individuals interact with their devices significantly impacts the accuracy of the model (Coskun et al., 2015). For instance, inertial and ambient sensor readings can vary depending on whether a device is statically mounted within a vehicle or carried by a traveler. In the latter case, natural movements such as those of the arms, wrists, or torso introduce additional variability to the sensor data. This highlights the importance of datasets that reflect the diversity of real-world usage scenarios, capturing a wide range of user behaviors to ensure robust and reliable model performance. Ultimately, the lack of diverse, high-quality, and representative datasets hinders development of robust, generalizable machine learning models for transport mode detection. The absence of standardized benchmarking datasets presents a significant challenge for comparing existing studies, as most rely on disparate datasets with varying characteristics (Aziz et al., 2024). Addressing these limitations is essential for advancing the field and enabling accurate, real-world applications of transport mode detection technology.

While sensor data is invaluable for understanding mobility patterns and transportation modes, it also raises substantial privacy concerns. Location data, for instance, can be cross-referenced with external datasets to identify individuals, potentially exposing sensitive information such as home and work addresses, personal movement patterns, and affiliations inferred from visits to specific venues (Huang et al., 2023). Although motion sensors are often regarded as privacy-friendly, research indicates they too can pose significant risks. Studies have shown that acceleration data, in particular, can enable serious privacy intrusions by facilitating inferences about a user's location, identity, demographics, personality traits, health status, emotions, or activities (Kröger et al., 2019). These privacy concerns have driven a growing interest in transport mode detection solutions that operate locally on users' devices, avoiding centralized data processing (Ferreira et al., 2020; Kamalian et al., 2022; Wang, Cao, et al., 2018). While on-device processing enhances privacy, it also introduces challenges related to computational efficiency, particularly given the resource constraints of smartphones.

This work addresses these challenges with the following contributions:

1. **Introduction of a Novel, Diverse Dataset (NOR-TMD):** We provide an openly available dataset

that addresses critical limitations in existing datasets, including both Android and iOS data, diverse devices and participants, and previously under-represented transportation modes such as boats.

2. **A Framework for Feature Evaluation and Reduction (EFT-TMD):** We present a comprehensive framework to identify and rank the most critical features, consisting of aggregated smartphone sensor data, enabling the development of smaller and more efficient models without compromising accuracy.
3. **Transferability of Identified Features:** The insights derived from our framework demonstrate the transferability of the identified features across diverse machine learning models, ensuring that the findings can be applied to various datasets and real-world scenarios.

We propose three key contributions to advance the field of transport mode detection. First, we introduce a novel and diverse dataset, NOR-TMD, addressing critical limitations in existing datasets by including iOS data, diverse devices, and under-represented transportation modes such as boats. NOR-TMD provides a valuable resource for developing and benchmarking generalizable transport mode detection models. Second, we present a systematic framework for feature evaluation and reduction (EFR-TMD), integrating multiple ranking techniques to identify the most critical features for transport mode detection. This framework facilitates the development of compact, efficient models that maintain high accuracy, making them suitable for real-time, on-device processing. Finally, we demonstrate the transferability of the identified features across diverse datasets, machine learning models, and real-world transportation contexts. Together, these contributions empower researchers and practitioners to design scalable, privacy-preserving, and generalizable solutions for intelligent transportation systems. In the following section, we review the existing literature with a focus on data sources, preprocessing techniques, and machine learning algorithms used in transport mode detection.

Related work

Significant research efforts have been dedicated to addressing the challenge of accurate transport mode detection, with much of this work leveraging machine learning approaches (Chen et al., 2017; Guvensan et al., 2017; Hemminki et al., 2013; Kamalian & Ferreira, 2022; Liang & Wang, 2017; Sankaran et al., 2014). Broadly speaking, methodologies in transport mode detection can be categorized into three main approaches: statistical methods, rule-based methods, and machine learning methods (Sadeghian et al., 2021). Statistical methods, while foundational, have demonstrated limitations in accurately distinguishing between similar transport modes, such as cars, buses, and trains. Similarly, rule-based approaches often struggle with differentiating between closely related modes of transportation. In contrast, machine learning techniques exhibit superior generalization capabilities, enabling more precise discrimination between similar modes of transport (Xiao et al., 2019). Most contemporary machine learning approaches in this domain leverage data collected from mobile devices to infer transportation modes. This process bears a strong resemblance to the field of human activity recognition, which also employs similar methods to deduce user activities rather than the mode of transportation. Human activity recognition is a key component of mobile computing, especially in regards to context awareness, and is the automated assessment of what a user is doing (Plötz & Guan, 2018). Traditionally, human activity recognition research has been rooted in computer vision. However, with the advent of ubiquitous technologies such as mobile devices and wearables, there has been a notable shift toward leveraging inertial sensors, such as accelerometers and gyroscopes, rather than image- and video-based data (Bulling et al., 2014). Transport mode detection can thus be viewed as a specialized subset of HAR, with many methodologies and insights from human activity recognition research being transferable to this domain. While both human activity recognition and transport mode detection predominantly employ machine learning techniques combined with mobile sensor data, there is significant variation in the types of data sources utilized, as well as in the preprocessing and feature extraction techniques employed for machine learning model training. In this section, we focus on three key aspects of transport mode detection: the types of data used as input, the techniques employed to process and prepare this data, and the machine learning algorithms applied to achieve accurate transport mode detection.

Data foundation

The data foundation for transport mode detection can generally be categorized into three main types: network data, geographical location data, and data from inertial and ambient sensors. Network data offers valuable information that can be leveraged to infer transportation modes. Coarse-grained network data, such as call detail records (CDR), provides approximate geographical location information and has been utilized in transport mode detection (Bachir et al., 2019). Other sources include GSM data and base station information (Asgari & Clemencon, 2018). Wireless access protocol (WAP) data, collected through wireless access points, has also shown potential when combined with inertial sensors. WAP data can identify nearby networks to approximate location, which can then be used to gather contextual details such as nearby public transport stations, stops, and services (Cardoso et al., 2016). While network information has proven useful, it is typically employed in combination with other data sources, such as inertial sensor data or geographical location data, to improve accuracy (Asgari & Clemencon, 2018; Bachir et al., 2019; Cardoso et al., 2016). Geographical location data, particularly GPS data, has been widely employed in TMD. Most studies utilize GPS in conjunction with other data sources to enhance differentiation between transport modes (Asgari & Clemencon, 2018; Bachir et al., 2019; Cavalcante et al., 2022; Kamalian & Ferreira, 2022; Nirmal et al., 2021). However, some research has demonstrated the feasibility of using GPS data alone to successfully distinguish between modes of transport (Sadeghian et al., 2022). When relying solely on GPS data, derived features such as acceleration, calculated from location data, are critical for determining the transport mode (Sadeghian et al., 2022). Inertial sensors, particularly accelerometers, are among the most commonly utilized data sources for transport mode detection and are key features in many transport mode detection systems (Cavalcante et al., 2022; Hemminki et al., 2013; Kamalian & Ferreira, 2022; Moreau et al., 2022; Nirmal et al., 2021). Modern smartphones are equipped with a variety of sensors, and there is considerable variation in the combinations of sensors used in research. Many approaches combine accelerometer data with other inertial sensors, such as magnetometers (Chen et al., 2017) and gyroscopes (Asci & Guvensan, 2019; Guvensan et al., 2017). Additional sensors, including rotation vectors and device orientation sensors, have also been incorporated in certain studies (Alotaibi, 2020; Moreau et al., 2022). Linear acceleration, gravity sensors, and barometers have also been utilized and have shown potential to improve accuracy (Alotaibi, 2020; Sankaran et al., 2014). The integration of data from multiple sensors generally results in higher accuracy for transport mode detection and according to Nikolic and Bierlaire (2017), leveraging a broader array of sensors increases the reliability and robustness of transportation mode classification.

Available datasets

While researchers often develop custom applications and recruit participants to collect sensor and location data for experimentation (Lorintiu & Vassilev, n.d.; Wang, Gao, et al., 2018; Xiao et al., 2015), these datasets are frequently not made publicly available, which poses a significant challenge for comparing results across studies (Aziz et al., 2024). Despite this limitation, several systematically collected datasets are publicly accessible. Notable examples include the HTC Transport Mode Dataset (Yu et al., 2014), the Sussex-Huawei Locomotion (SHL) Dataset (Gjoreski et al., 2018), the US-TMD Dataset (Carpineti et al., 2018), and the Collecty Dataset (Erdelić et al., 2023). The HTC Transport Mode Dataset comprises 8,311 h of data collected from 150 students and 74 employees or interns, totaling 100 GB (Yu et al., 2014). Although extensive in terms of participant count, data volume, and transportation modes, its geographic diversity and generalizability are constrained due to data collection along only two predefined routes. Additionally, the dataset lacks diversity in sensors and devices. In contrast, the SHL Dataset, collected over seven months by three participants, contains 950 GB of data spanning 2,812 h (Gjoreski et al., 2018). This dataset includes data from 15 different smartphone sensors, offering greater sensor diversity than the HTC dataset. However, it was collected using a single device type, which may limit its generalizability despite its larger scale and broader range of sensor modalities. The US-TMD Dataset includes approximately 32 h of data (3 GB) collected from 13 participants using 11 different device types (Carpineti et al., 2018). While smaller in scale compared to the HTC and SHL datasets, its inclusion of multiple device types enhances its diversity and applicability across different hardware configurations. Similarly, the Collecty Dataset contains approximately 242 h of data collected from 15 participants over five months (Erdelić et al., 2023). Though smaller in size than the SHL Dataset, it offers additional value by contributing to the pool of publicly available datasets with a balanced focus on user diversity and extended data collection periods. Diversity plays a critical role in creating models that generalize across environments (Gong et al., 2019). However, existing datasets often lack diversity in key areas such as participants, devices, sensors, and geographic regions, limiting their generalizability. This highlights the pressing need for more comprehensive datasets to address these gaps and enhance model robustness in real-world scenarios.

Feature extraction and preprocessing

While a wide range of data sources can be utilized for achieving accurate transport mode detection, there is an equally broad variation in the methods and techniques employed to extract useful features and prepare these data sources for training machine learning models. Bulling et al. (Bulling et al., 2014) introduced the Activity Recognition Chain (ARC) framework, which outlines the sequence of steps required to infer a user's activity using inertial sensors in the context of human activity recognition. A key aspect of the framework is segmentation, for which the authors recommend the use of a sliding window over time-series data to extract meaningful segments. Although the ARC framework primarily targets activity recognition, the sliding window approach has been widely adopted in transport mode detection as well (Burkhard et al., 2020; Cardoso et al., 2016; Widhalm et al., 2018). The length of these sliding windows typically ranges from one to 60 s. Longer windows may also be employed, however, there is a tradeoff between achieving finer granularity and reducing computational resource requirements, as well as maintaining higher window density (Matthes & Springer, 2022). For three-dimensional inertial sensor data, many studies compute the magnitude of the three axes to eliminate directional dependencies before aggregating the data segments (Cardoso et al., 2016; Widhalm et al., 2018; Yu et al., 2014). Similarly, to address time dependencies, some researchers transform the time-series data into frequency data using Fourier transforms (Bulling et al., 2014; Matthes & Springer, 2022). A variety of aggregation functions have been employed to summarize segmented data and to define features for transport mode detection. Statistical functions such as variance (Widhalm et al., 2018), average (Asgari & Clemencon, 2018; Cardoso et al., 2016; Kamalian & Ferreira, 2022; Sadeghian et al., 2022), median (Cardoso et al., 2016), minimum and maximum values (Cardoso et al., 2016; Kamalian & Ferreira, 2022; Sadeghian et al., 2022), kurtosis (Cardoso et al., 2016), standard deviation (Kamalian & Ferreira, 2022), percentiles (Das & Winter, 2018), quartiles (Burkhard et al., 2020), and interquartile range (Nirmal et al., 2021) have all been used to extract meaningful features for machine learning models. For smartphone-based data collection, the majority of research relies on Android sensor data (Das & Winter, 2018; Kamalian & Ferreira, 2022; Nirmal et al., 2021; Widhalm et al., 2018), with relatively few studies incorporating iOS sensor data in their proposed solutions or evaluations (Asgari & Clemencon, 2018). Despite this, there remains significant diversity in preprocessing techniques and feature selection methods used to prepare data for machine learning models. The variety of algorithms employed for these purposes further reflects the breadth of approaches explored in transport mode detection research.

Machine learning algorithms

A wide range of machine learning algorithms have been utilized to infer modes of transportation, spanning the majority of fundamental algorithmic classes (Nikolic & Bierlaire, 2017). These include decision trees (Cardoso et al., 2016; Ferreira et al., 2020), random forests (Kamalian & Ferreira, 2022), k-nearest neighbors (Sadeghian et al., 2022), Bayesian inference (Bachir et al., 2019), hidden Markov models (Widhalm et al., 2012) and support vector machines (Nikolic & Bierlaire, 2017). In addition to foundational methods, more advanced sub-classes of algorithms have been utilized, including adaptive boosting (AdaBoost) (Muharemi et al., 2020) and extreme gradient boosting (XGBoost) (Lu et al., 2019). Furthermore, neural network architectures such as multilayer perceptrons (MLPs) (Mastalerz et al., 2020), recurrent neural networks (RNNs) (Vu et al., 2016), and convolutional neural networks (CNNs) (Liang & Wang, 2017) have also been explored for transport mode detection. In transport mode detection systems, the placement of the model within the system architecture significantly impacts factors such as resource consumption, privacy, and inference speed. Common deployment strategies involve server-side placement (Nirmal et al., 2021) or on-device placement (Cavalcante et al., 2022). Some studies further explore distributed architectures, such as fog computing, where inference models are deployed on fog nodes to balance computational efficiency and latency (Kamalian & Ferreira, 2022). The range of transport modes that these solutions can infer varies widely across studies. While some approaches target a minimal set of three modes (Cavalcante et al., 2022), others incorporate a broader spectrum of transportation options (Asgari & Clemencon, 2018; Cardoso et al., 2016; Kamalian & Ferreira, 2022). Commonly inferred modes include bus, train, car, metro, bicycle, and walking. A few studies extend their scope to include less commonly addressed modes, such as seagoing vessels (Guvensan et al., 2017), though such efforts remain rare.

Summary of past research

The current state of transport mode detection research underscores significant reliance on sensor data, such as accelerometers, gyroscopes, magnetometers, and barometers, for distinguishing transportation modes. However, publicly available datasets often lack diversity in terms of devices, operating systems, and transportation modes, limiting their generalizability and usefulness for benchmarking. To our knowledge, there are no systematic frameworks for evaluating feature importance, despite the widespread use of varied preprocessing techniques, aggregation functions, and machine learning approaches. This highlights the need for more diverse datasets, a structured approach to identifying critical features, and deeper insights into the relative importance of sensor data features commonly employed in transport mode detection research. Addressing these gaps is essential for developing robust, generalizable, and efficient transport mode detection solutions.

Dataset

To address the challenges and limitations identified in existing publicly available datasets, we present a novel dataset, *NOR-TMD*, specifically designed to enhance the generalizability, diversity, and applicability of transport mode detection research. NOR-TMD addresses these limitations by increasing device and sensor diversity, expanding participant counts, and incorporating transportation modes and sensors not previously represented. Furthermore, unlike other datasets, it includes data collected from both iOS and Android devices, ensuring broader platform coverage and applicability. By systematically collecting data from a wide range of participants, diverse geographic locations, and multiple device types, NOR-TMD provides a more comprehensive foundation for developing and evaluating transport mode detection methods. This section describes the data collection methodology, provides a comprehensive overview of the NOR-TMD dataset, and outlines the preprocessing steps implemented to clean and refine the raw sensor data.

Data collection

To populate NOR-TMD, we recruited 101 regular travelers, including men and women across diverse age groups and occupations. Recruitment was conducted in two cities in Norway through multiple communication channels. Data collection was facilitated using a custom mobile application developed for both iOS and Android platforms. For detailed information about the application, we refer interested readers to our previous work (Skretting et al., 2024). The application was designed to activate and record events from all available sensors on a device at a sampling frequency of 5 Hz (200 ms). Due to variations in operating system resource management, hardware differences, and sensor prioritization, the actual sampling frequency varied. Nevertheless, a high frequency was prioritized to maximize data collection, with the option to adjust the frequency later using interpolation techniques. Participants manually initiated data collection by selecting a transportation mode from a predefined list and stopped recording upon disembarking. Data was stored locally on the device, and after each session, participants were prompted to review a summary of their trips, including the mode of transportation, date, time, and trip duration. They were required to verify the accuracy of the labels before uploading the data to the server. To ensure high-quality data collection, all participants attended a mandatory two-hour, in-person training session. This session emphasized the importance of accurate labeling and instructed participants to delete any data they were uncertain about. Participants were divided into small groups and accompanied by instructors on public transport trips for hands-on guidance. Data collection spanned one month, during which participants were incentivized with free public transport tickets. Email support was provided throughout the period, and any mislabeled or incorrectly uploaded data was promptly deleted from the database. These measures minimized errors and ensured high-quality data collection. Figure 1 presents the distribution of devices used in the study.

Sensor removal and data cleaning

The dataset consists of raw sensor data collected from a variety of smartphone sensors on both iOS and Android platforms. Several preprocessing steps were implemented to ensure the data's quality and relevance for transport mode detection solutions. Out of the wide variety of different recorded smartphone sensors, manufacturer-specific sensors (e.g., those unique to LG, Samsung, and Xiaomi devices) were excluded to enhance the model's generalizability, focusing instead on standard sensors available across operating systems. Hardware-specific sensors unrelated to transport mode detection, such as those linked to camera or screen functionality and device orientation, were

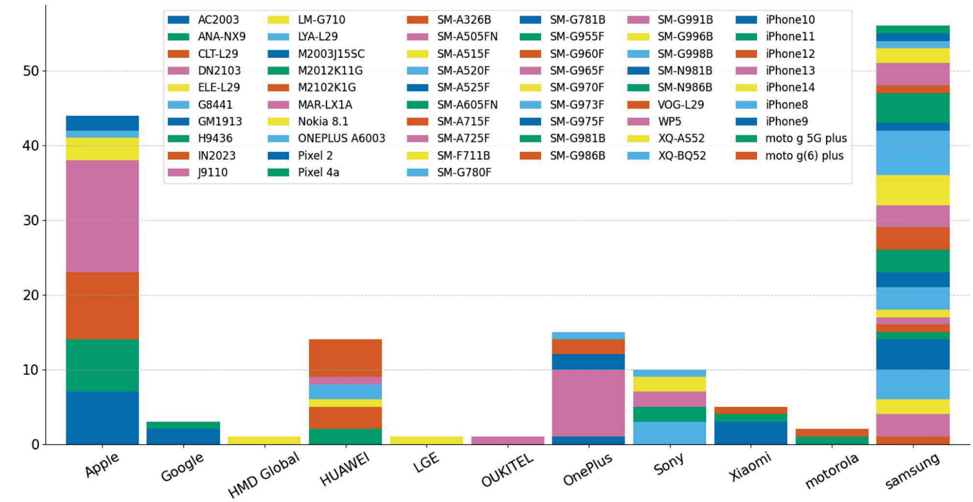


Figure 1. Distribution of devices used to collect nor-TMD.

also omitted. Additionally, undocumented or nonstandard Android sensors were excluded due to their limited applicability and the absence of sufficient documentation. Sensors with fewer than 100,000 recorded entries during the data collection period were removed because their low data density could hinder efficient inference. Sensors irrelevant to transport mode detection, including proximity sensors (which measure the distance to external objects), step counters, and ambient light sensors, were also excluded. To further refine the dataset, sensor events associated with trips exceeding one hour in duration were discarded, as such trips might reflect instances where users forgot to stop the data collection process. Additionally, trips lasting less than 60 s were removed, as such short durations could indicate instances where participants accidentally started data collection. The final set of sensors included in the dataset is detailed in Table 1. Figures 2 and 3 illustrate the distribution of data across different transportation modes for Android and iOS devices, respectively.

Table 1. Number of recorded sensor events.

Android	Events	iOS	Events
Accelerometer	71 466 000	Gravity	3 859 042
Gyroscope	53 280 748	Rotation rate	3 859 042
Gyroscope unc.	42 118 113	Quaternion	3 859 042
Magnetic field	34 138 971	Gyroscope	3 857 737
Rotation vector	31 602 332	Accelerometer	3 857 017
Game rotation vector	29 521 508	Magnetic field	3 856 122
Linear acceleration	29 464 998	Pressure	759 027
Gravity	29 453 767	Activity recognition	151 154
Geomagnetic rot. Vector	19 964 410		
Magnetic field unc.	18 021 641		
Pressure	4 369 367		
Activity recognition	4862		

Dataset composition

The data collection and cleaning processes resulted in a total of 387 million unique sensor events, comprising 363 million events from Android devices and 24 million events from iOS devices. The substantial disparity in data volume between the two platforms can be attributed to three key factors. First, Android devices offer a greater variety of sensor types compared to iOS devices, which have access to a more limited set of native sensors. Second, the study included approximately twice as many Android users as iOS users, contributing to the larger data volume from Android devices. Finally, although we specified the same sampling rate for both platforms, the Android operating system of the devices included in the study seemingly supports higher sampling rates, further amplifying the data volume difference. Sensor readings were collected within one of the predefined transportation modes: *bicycle*, *boat*, *bus*, *car*, *e-scooter*, *inside*, *metro*, *outside*, *train*, and *tram*. While most of these transportation modes are self-explanatory, the modes “inside” and “outside” warrant additional clarification. The “inside” label refers to instances where travelers were situated inside a building, rather than within a vehicle. Similarly, the “outside” label corresponds to cases where individuals were outdoors but not in a

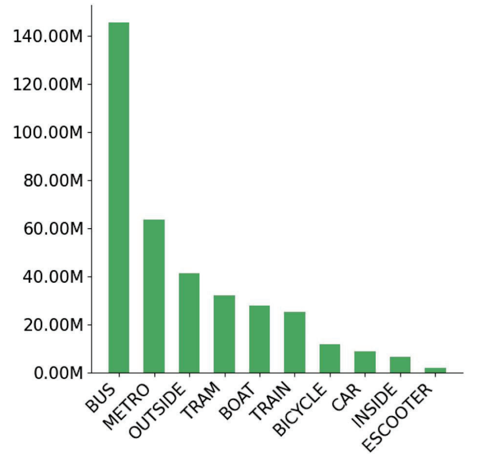


Figure 2. Distribution of Android sensor data across different transport modes.

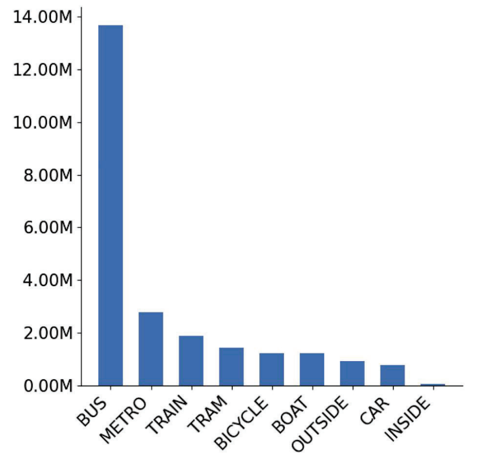


Figure 3. Distribution of iOS sensor data across different transport modes.

transportation vehicle, for instance, walking or standing outside. These additional labels were included to enable comparisons with similar transportation modes and to identify potential dissimilarities. For example, training a machine learning model using ambient sensor data to distinguish between bicycles and e-scooters might inadvertently result in a model that primarily detects whether an individual is outdoors, as many sensor readings associated with these modes are influenced by the outdoor environment. Similarly, data labeled as "inside" could exhibit characteristics that overlap with those of being inside a vehicle, such as a train. Capturing these modes allows for a more nuanced analysis and helps prevent such confounding effects in the training and evaluation of transport mode detection models. This dataset has been made publicly available on Kaggle (Skretting & Grønli, 2025).

Feature evaluation framework

In this section, we introduce our framework for feature evaluation, EFR-TMD (Ensemble Feature Ranking for Transport Mode Detection). EFR-TMD is designed to identify the smartphone sensors and statistical aggregations that are most advantageous for transport mode detection. By pinpointing the most impactful features, the framework enables the elimination of less informative ones, thereby reducing training time, inference time, and model size. This optimization ensures that models maintain high accuracy while operating more efficiently, particularly on resource-constrained devices.

Conceptualization

There are numerous techniques available for ranking feature importance, many of which are tailored to specific algorithms. However, some generic methods exist, such as Permutation Importance (PI) (Altmann et al., 2010), Shapley Additive Explanations (SHAP) (Lundberg & Lee, 2017), Mutual Information (MI) (Battiti, 1994), the Analysis of Variance (ANOVA) F-test (Ståhle & Wold, 1989), and Classification and Regression Trees (CART) (Breiman, 2017). Permutation Importance is a technique used in machine learning to assess feature importance by randomly shuffling the values of a feature and evaluating the impact on model performance. The performance drop after shuffling reflects the importance of the feature, with the average performance decrease across multiple iterations used to rank features. Mutual Information quantifies the dependency between a feature and the target variable by measuring their statistical dependence, thus revealing how much information a feature contains about the target variable. The ANOVA F-test is a statistical method that compares the means of two or more groups to assess whether they differ significantly. It calculates an F-score that can be used to rank features based on their contribution to the target variable. SHAP, on the other hand, adopts a game-theoretic approach to explain the output of any machine learning model by distributing contributions fairly among features, providing interpretable feature importance scores. Finally, CART-based feature importance evaluates how

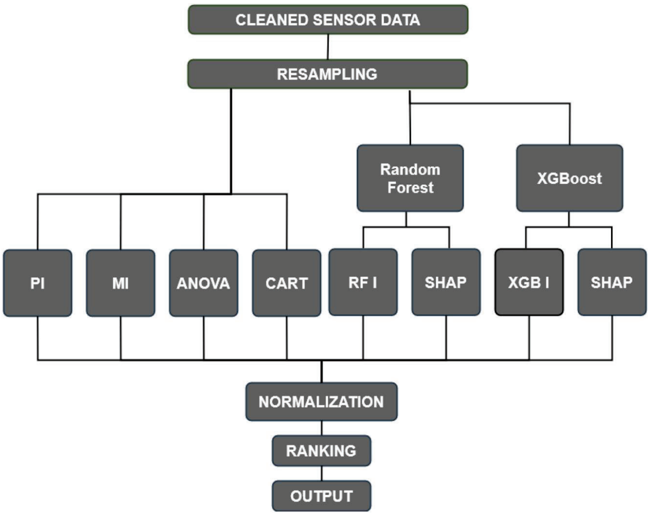


Figure 4. High-level overview of EFR-TMD.

much each feature contributes to the decision-making process during the construction of decision trees. While these methods are widely used, they often produce significantly different results for the same dataset. This variability makes it challenging to determine which method is most appropriate for a specific context and whether the results are reliable. To address these challenges, we propose a comprehensive framework that integrates all these methods, alongside the built-in feature importance scores available from Random Forest and XGBoost models. In our framework, each feature set is evaluated independently by these techniques, and the resulting scores are consolidated and normalized to produce a unified importance score for each feature. To normalize the data, the output from each importance method is sorted in descending order based on the feature scores, creating an ordered list for each method. These ordered lists are then consolidated into a single list by averaging the rank positions of each feature across all methods. Since each feature is defined as a combination of a sensor and an aggregation function, the ranking of individual sensors and aggregation functions is determined by calculating the average rank across all features that include the same sensor or function. The conceptual flow of the framework is illustrated in Figure 4.

Ranking features using the proposed framework

Now that we have presented our framework, we apply it to NOR-TMD to evaluate the importance scores of commonly used sensor values and aggregation functions. Before the data can be processed by the framework, certain preprocessing steps are performed to ensure data quality and consistency. This analysis aims to identify the most influential features, providing insights into which sensors and aggregation methods contribute most effectively to transport mode detection.

Preprocessing

Before applying the framework to derive feature importance, we performed preprocessing steps to prepare the sensor data for analysis. We applied the consecutive processing steps on all sensor data from NOR-TMD, except the built-in activity recognition. While various data processing techniques, such as applying Gaussian filters (Chen et al., 2017) or smoothing (Liang & Wang, 2017) can be applied to sensor values, we opted to retain the data as close to its raw form as possible to evaluate all sensors in a uniform and general manner. Since the goal of this study is to highlight the importance of sensors generically, we eliminated directional dependencies by calculating the magnitude of the directional axes (x, y, and z) for each sensor, defined as $\sqrt{x^2 + y^2 + z^2}$. To segment the time-series data, we employed a window function. As noted by Matthes and Springer (2022), the choice of window length significantly impacts inference time, resource consumption, and model accuracy. Drawing on previous studies (Guvensan et al., 2017; Liang & Wang, 2017; Matthes & Springer, 2022; Oplenskedal et al., 2021), we adopted a

Table 2. Statistical aggregations applied to each sensor.

Aggregation function	Description
Minimum	Smallest value in the segment
Maximum	Largest value in the segment
Average	Mean value in the segment
Range	Difference between maximum and minimum
Variance	Variability in the segment
Standard deviation	Dispersion around the mean
Kurtosis	Measure of data distribution shape
1st quartile	25th percentile of the data
2nd quartile (median)	50th percentile of the data
3rd quartile	75th percentile of the data
Interquartile range	Range between 1st and 3rd quartiles

10-second window function with a 5-s overlap, given its demonstrated effectiveness in prior research. For each segment, we aggregated the magnitude data using a diverse set of statistical functions derived from the literature. These preprocessing steps were uniformly applied to both the iOS and Android datasets. While the collected dataset is extensive and captures a range of transportation modes, it is inherently skewed due to participants using their regular means of transportation during the data collection period. This imbalance is evident in Figures 2 and 3, where significantly more data was collected onboard buses compared to other transportation modes. The disparity among classes allows us to categorize them into *majority classes* and *minority classes*. Such imbalances can pose challenges, as classifiers may become biased toward majority classes (Kaur et al., 2019). To address this issue, we employed resampling techniques to reduce dataset skewness. Resampling methods have proven effective in supervised learning, often outperforming bagging and boosting techniques (Kaur et al., 2019). In this study, we utilized SMOTE (Synthetic Minority Over-sampling Technique) (Chawla et al., 2002) due to its widespread adoption and demonstrated robustness in the research community (Chen et al., 2017; Matthes & Springer, 2022). Prior to resampling, we re-labeled certain classes with significantly less data (e.g., *inside*, *outside*, and *e-scooter*) into a single composite class, *other*. This composite class serves as a control, distinguishing instances where travelers are not aboard public transportation. Both the majority class (*bus*) and the composite class (*other*) were left unchanged. The dataset was then split into training (66%) and test (33%) sets. We oversampled the minority classes (*bicycle*, *boat*, *car*, *metro*, *train*, and *tram*) in the training set by 20%. The resulting features consist of each sensor aggregated with the aforementioned statistical functions. The final datasets comprise eight transportation classes and 121 features for Android and 77 features for iOS. These features were derived by applying the statistical aggregations listed in Table 2 to the magnitude data all the sensors shown in Table 1, except activity recognition. For example, one feature represents the maximum acceleration within a window segment, while another captures the average magnetic field calculated over the same segment. To incorporate data from pressure sensors in our evaluation, we limited the analysis to devices equipped with this sensor, significantly reducing the dataset size.

Additionally, the use of a 10-second window function aggregated all sensor values within each window into a single value per feature, further decreasing the data points. The final dataset for analysis comprised 130,405 rows of data: 39,974 from Android and 90,431 from iOS devices. The disproportionate representation of iOS data stems from the prevalence of air pressure sensors in iOS devices, whereas many Android devices lack this sensor.

Feature ranking for android

To rank the importance of features in the Android dataset, we applied the previously described framework to all 121 features described in Tables 1 and 2.

Figures 5 and 6 display the average rank for each sensor and aggregation function, respectively, as evaluated by each importance method included in the proposed framework. Figures 7 and 8 provide an aggregated view of the average rank for sensors and functions across all methods, while Figure 9 shows the top 30 ranked features (sensor-function combinations). From the results, we observe that acceleration, pressure, and magnetic field are the three highest-ranked sensors, while range, interquartile range, and standard deviation are among the most influential aggregation functions. These findings highlight the critical sensors and statistical functions that contribute most effectively to transport mode detection on Android devices.

Feature ranking for iOS

To rank the importance of features in the iOS dataset, we applied the previously described framework to all 77 features derived from the combination of sensors and functions described in Tables 1 and 2. Figures 10 and 13 display the average rank for each sensor and aggregation function, respectively, as evaluated by each importance method in the framework. Figures 11 and 12 provide an aggregated view of the average rank for sensors and functions across all methods, while Figure 14 shows the top 30 ranked features (sensor-function combinations).

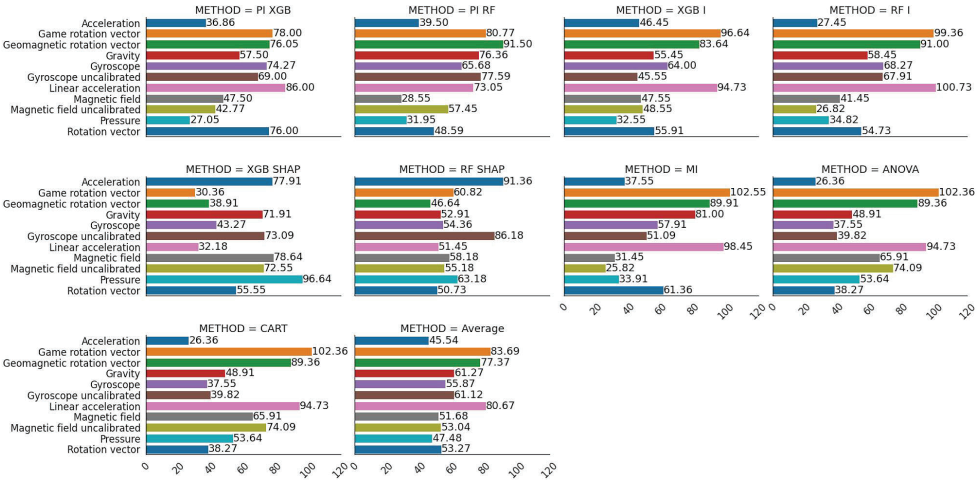


Figure 5. Average rank of each sensor for each of the feature importance methods using the Android dataset.

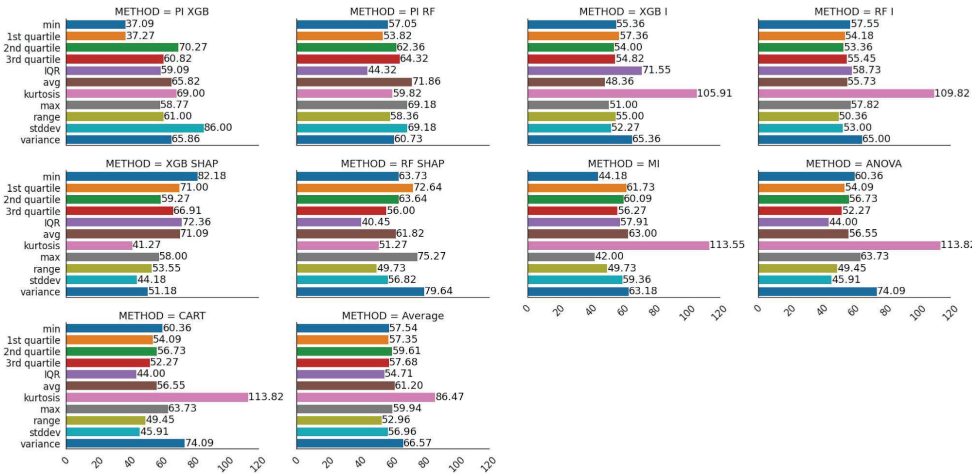


Figure 6. Average rank of each function for each of the feature importance methods using the Android dataset.

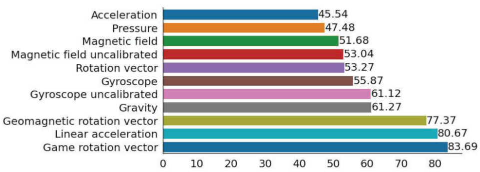


Figure 7. Average rank of each sensor across all feature importance methods using the Android dataset.

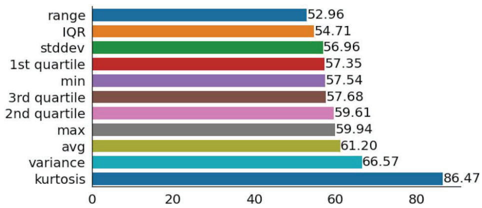


Figure 8. Average rank of each function across all feature importance methods using the Android dataset.

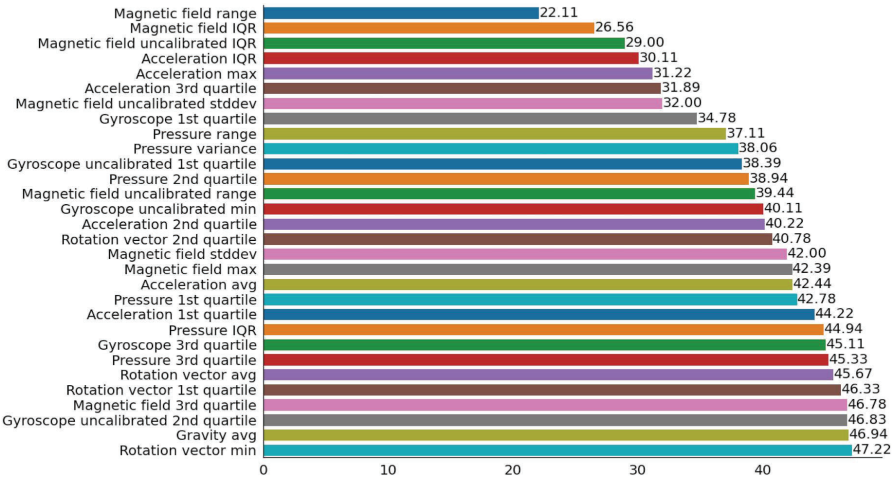


Figure 9. Average rank of each composite feature across all feature importance methods using the Android dataset.

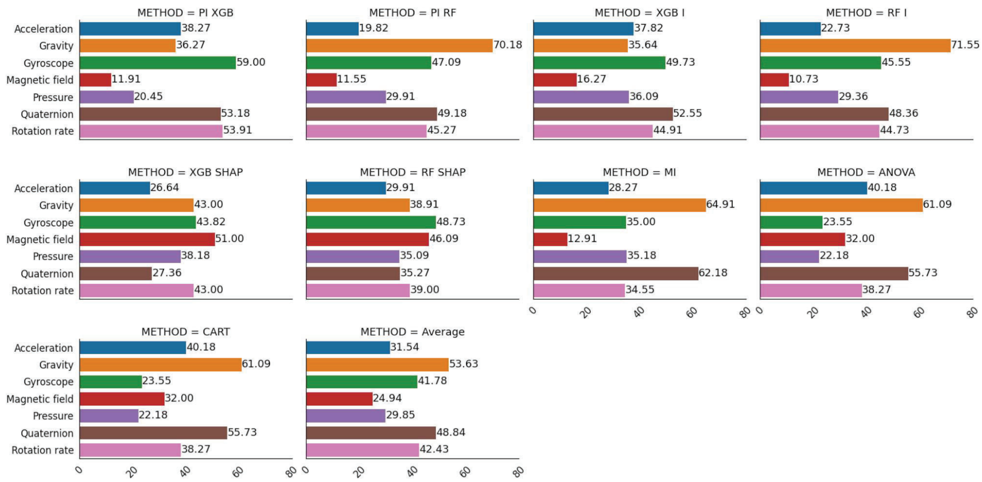


Figure 10. Average rank of each sensor for each of the feature importance methods using the iOS dataset.

From the results, we observe that magnetic field, pressure, and acceleration are the three highest-ranked sensors, consistent with the Android dataset. However, while interquartile range remains a top-ranked aggregation function, the rankings of range and standard deviation are lower compared to Android. These findings underscore both similarities and platform-specific differences in feature importance for TMD.

Evaluation of framework

In the preceding section, we ranked the importance and impact of various sensors and aggregation functions for transport mode detection. This section evaluates the practicality and effectiveness of the proposed framework, EFR-TMD by examining how the removal of low-importance features, as identified by the framework, affects model performance. Specifically, we train classifiers using the complete set of available features and then iteratively remove 25% of the features, starting with those ranked lowest in importance. We evaluate the changes in training time, inference time, and classification accuracy at each step of feature removal. The evaluation includes a diverse set of classification algorithms: convolutional neural networks (CNNs), multilayer perceptrons (MLPs), long short-term memory networks (LSTMs), Random Forest (RF), and XGBoost (XGB). The focus of this evaluation is not on optimizing the models themselves but on assessing the impact of the feature reduction process facilitated by EFR-TMD. To ensure consistency, all neural network models were implemented with a basic architecture. Each model consisted of a single input layer with 32 nodes/filters and a single output layer. The rectified linear unit (ReLU) activation function was applied to the input layer, while the softmax activation function was used in the output layer for multi-class classification. All models were compiled using the Adam optimizer and categorical cross-entropy as the loss function. Training was conducted over 150 epochs for each model. We trained models separately on the Android and iOS subsets of the identified features from the NOR-TMD dataset. The training dataset, consisting of 66% of the data, was employed to train the models, while the test set, consisting of 33% of the data, was utilized to evaluate the models. To enable effective training across all model architectures, we employed a MinMaxScaler to normalize the features by linearly scaling them within a fixed range. This widely recognized approach has been shown to improve model performance and stability, outperforming other scaling techniques in previous studies (Raju et al., 2020).

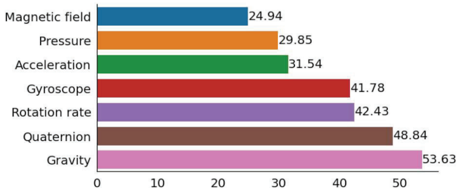


Figure 11. Average rank of each sensor across all feature importance methods using the iOS dataset.

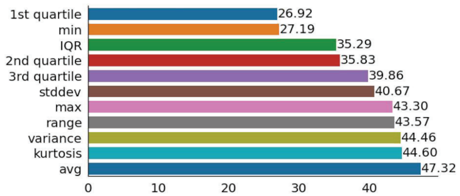


Figure 12. Average rank of each function across all feature importance methods using the iOS dataset.

Android

The baseline models trained on the Android data utilized all 121 features, establishing a benchmark to represent model performance without applying any feature reduction.

Two distinct approaches were employed for feature selection: one based on individually ranked sensors and aggregation functions, and another using a ranked list of composite features. For the sensor-function ranking approach, a matrix was constructed comprising all possible combinations of sensors and aggregation functions (121 features in total). Feature subsets were created containing approximately 75%, 50%, and

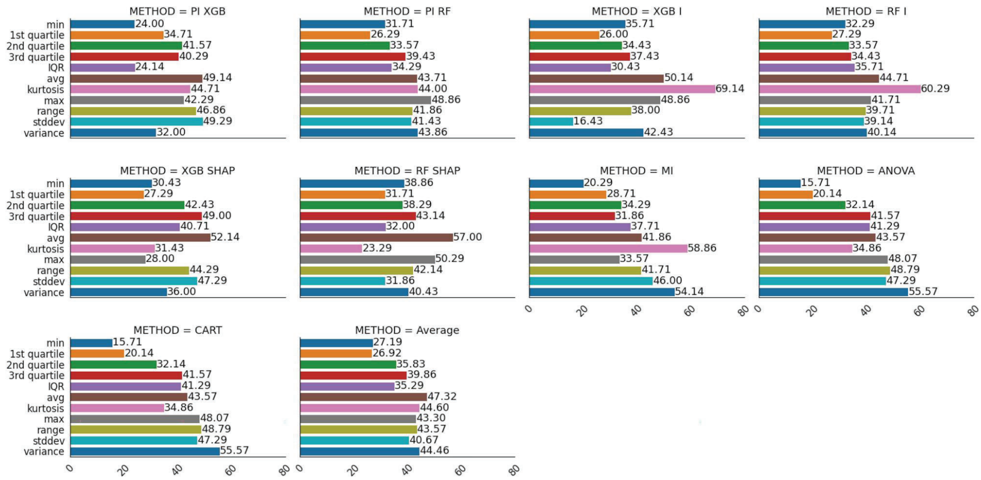


Figure 13. Average rank of each function for each of the feature importance methods using the iOS dataset.

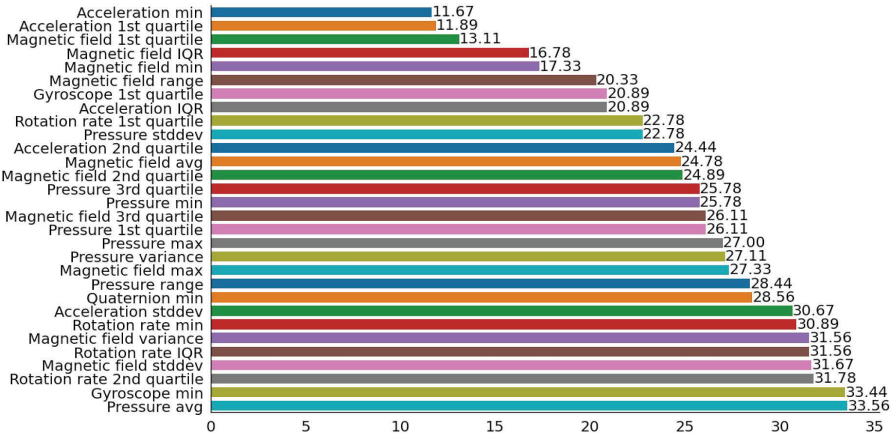


Figure 14. Average rank of each composite feature across all feature importance methods using the iOS dataset.

Table 3. Sensor and function ranking evaluation on Android.

Algorithm	# Features	Inference time (ms)	Training time (m)	Accuracy (%)
CNN (100%)	121	55.56	5.38	93.47
CNN (75%)	81	45.96	4.59	93.43
CNN (50%)	36	46.65	3.65	90.60
CNN (25%)	9	45.31	2.80	80.31
LSTM (100%)	121	47.32	69.36	93.77
LSTM (75%)	81	48.24	46.55	93.74
LSTM (50%)	36	47.58	21.25	93.84
LSTM (25%)	9	46.45	8.34	82.78
MLP (100%)	121	47.03	3.19	93.03
MLP (75%)	81	45.63	3.19	93.94
MLP (50%)	36	46.02	2.92	90.92
MLP (25%)	9	46.24	2.54	80.63
RF (100%)	121	9.34	1.15	96.55
RF (75%)	81	8.86	0.83	96.78
RF (50%)	36	8.23	0.57	95.13
RF (25%)	9	7.63	0.27	83.20
XGB (100%)	121	1.21	0.48	98.28
XGB (75%)	81	1.20	0.32	98.28
XGB (50%)	36	0.98	0.14	97.42
XGB (25%)	9	0.66	0.05	84.10

25% of the highest-ranked sensors combined with the 75%, 50%, or 25% highest-ranked aggregation functions. As 75%, 50%, or 25% of 11 sensors or aggregation functions do not result in whole numbers, these values were rounded up. These feature subsets were used to train models (CNN, MLP, LSTM, XGBoost, and Random Forest) to classify eight transportation modes: *bicycle*, *boat*, *bus*, *car*, *metro*, *train*, *tram*, and a composite *other* class. For each model, we recorded the training time, classification accuracy on unseen data, and the average inference time across 10,000 predictions. The results for this approach are presented in Table 3. For the composite feature ranking approach, three additional feature sets were created, containing 75%, 50%, and 25% of the top-ranked composite features. These subsets were also evaluated against the benchmark, with results presented in Table 4.

iOS

We applied the same two approaches for feature ranking to the iOS dataset as were used for the Android data. In the first approach, feature subsets were constructed using 75%, 50%, and 25% of the highest-ranked individual sensors combined with 75%, 50%, and 25% of the highest-ranked aggregation functions, respectively. As with the Android dataset, the number of sensors and functions was rounded up to the nearest integer. Models (MLP, CNN, LSTM, XGBoost, and Random Forest) were trained using these feature subsets and compared against the benchmark, which consisted of all 77 features.

In the second approach, subsets were created from the 75%, 50%, and 25% highest-ranked composite features. For each model, we recorded the training time, classification accuracy on unseen data, and the average inference time across 10,000 predictions. Training and inference times are reported in seconds, and accuracy is presented as a percentage. The results for models trained with subsets based on individual sensor and aggregation function rankings are presented in Table 5, while those for subsets based on composite features are shown in Table 6.

Discussion

This study introduced a novel dataset, NOR-TMD, along with a systematic framework for feature evaluation and reduction in transport mode detection

Table 4. Composite feature ranking evaluation Android.

Algorithm	# Features	Inference time (ms)	Training time (m)	Accuracy (%)
CNN (100%)	121	55.56	5.38	93.47
CNN (75%)	91	49.25	4.37	94.5
CNN (50%)	61	52.17	4.38	92.63
CNN (25%)	31	49.64	3.22	90.17
LSTM (100%)	121	47.32	69.36	93.77
LSTM (75%)	91	59.03	58.75	90.17
LSTM (50%)	61	52.34	41.07	91.19
LSTM (25%)	31	50.15	20.6	90.25
MLP (100%)	121	47.03	3.19	93.03
MLP (75%)	91	55.26	3.19	94.15
MLP (50%)	61	50.66	2.93	92.5
MLP (25%)	31	47.7	2.66	90.27
RF (100%)	121	9.34	1.15	96.55
RF (75%)	91	8.87	0.86	96.57
RF (50%)	61	8.34	0.66	96.16
RF (25%)	31	7.9	0.45	95.41
XGB (100%)	121	1.21	0.48	98.28
XGB (75%)	91	1.56	0.42	98.33
XGB (50%)	61	2.35	0.57	98.1
XGB (25%)	31	0.71	0.1	97.16

Table 5. Sensor and function ranking evaluation iOS.

Algorithm	# Features	Inference time (ms)	Training time (m)	Accuracy (%)
CNN (100%)	77	48.64	9.35	77.49
CNN (75%)	54	52.89	8.78	79.11
CNN (50%)	24	49.47	7.73	76.73
CNN (25%)	6	46.7	6.32	67.09
LSTM (100%)	77	53.86	113.13	81.38
LSTM (75%)	54	54.26	163.2	87.77
LSTM (50%)	24	50.07	41.1	81.35
LSTM (25%)	6	46.67	16.02	72.28
MLP (100%)	77	51.22	7.37	79.51
MLP (75%)	54	51.69	7	77.14
MLP (50%)	24	48.49	6.46	76.59
MLP (25%)	6	44.94	6.37	67.07
RF (100%)	77	9.29	1.85	90.54
RF (75%)	54	8.57	1.63	89.08
RF (50%)	24	8.16	0.88	88.72
RF (25%)	6	7.69	0.42	81.76
XGB (100%)	77	1.3	0.62	93.06
XGB (75%)	54	1.4	0.53	91.11
XGB (50%)	24	1.12	0.24	89.47
XGB (25%)	6	0.81	0.1	78.5

systems, EFR-TMD. By leveraging data collected from both Android and iOS devices, the framework successfully identified the most impactful sensors and aggregation functions. The evaluation demonstrated that feature reduction guided by our framework significantly reduces computational costs while maintaining competitive accuracy, making it particularly suitable for real-time and on-device processing. In this section, we critically compare NOR-TMD with existing datasets and discuss its practical applicability, along with the proposed framework and the insights derived from it.

Dataset comparison

We have proposed a new dataset to address some of the limitations of existing datasets, including lack of geographical, sensor, and device diversity. NOR-TMD exhibits a wider variety of transportation modes, as well as including annotations as to the placement of the device during data collection (e.g., hand or pocket). NOR-TMD is collected by regular travelers during their normal commuting process, making the collected data as realistic as possible. Maybe more importantly, NOR-TMD includes data collected on iOS devices, which none of the existing publicly available datasets contain. As previously mentioned, the main datasets that exist, including smartphone sensor data captured in different modes of transportation are the HTC Transport Mode dataset (Yu et al., 2014), Sussex-Huawei Locomotion (SHL) dataset (Gjoreski et al., 2018), the US-TMD dataset (Carpineti et al., 2018) and the Collecty dataset (Erdelić et al., 2023). Table 7 exhibits the main differences between these preexisting datasets and our proposed dataset. The development of NOR-TMD was motivated by limitations in existing publicly available datasets for transport mode detection, such as the HTC, SHL, US-TMD, and Collecty, datasets. Each of these datasets contributes valuable insights but exhibits significant constraints in terms of device diversity, participant representation, and transportation mode coverage, which hinder their generalizability and practical applicability. The HTC dataset stands out for its scale, with 8,311 h of data collected from 224 participants, yet it is restricted to a single device type. This lack of device diversity poses a challenge since sensor data quality and variability differ across devices due to differences in hardware, operating systems, and manufacturer-specific sensor implementations (Lane et al., 2010).

Similarly, the SHL dataset, while including data from 15 sensors, suffers from limited participant

Table 6. Composite feature ranking evaluation iOS.

Algorithm	# Features	Inference time (ms)	Training time (m)	Accuracy (%)
CNN (100%)	77	48.64	9.35	77.49
CNN (75%)	58	53.97	9.37	78.15
CNN (50%)	39	50.65	8.38	78.81
CNN (25%)	20	43.43	6.74	77.31
LSTM (100%)	77	53.86	113.13	81.38
LSTM (75%)	58	56.8	88.93	82.4
LSTM (50%)	39	49.49	62.6	83.24
LSTM (25%)	20	43.42	31.41	80.6
MLP (100%)	77	51.22	7.37	79.51
MLP (75%)	58	53.41	7.37	78.55
MLP (50%)	39	46.25	7.37	77.96
MLP (25%)	20	43.54	5.91	76.76
RF (100%)	77	9.29	1.85	90.54
RF (75%)	58	8.7	1.59	89.87
RF (50%)	39	8.14	1.33	88.8
RF (25%)	20	7.73	0.83	89.08
XGB (100%)	77	1.3	0.62	93.06
XGB (75%)	58	1.42	0.56	92.35
XGB (50%)	39	1.18	0.33	90.79
XGB (25%)	20	0.83	0.17	88.86

Table 7. Overview of publicly available sensor-based datasets.

Dataset	Modes	Sensors	OS	DL	N.UD	N.UP	Hours
HTC	S, W, R, BI, MC, C, B, M, TR	A, M, G	Android	N/A	1	224	8311 h
SHL	S, W, R, BI, C, B, M, TR	A, M, G, O, GR, L, P, AR, AL, BL, BT, WR, SR, CR, GPS, AU	Android	Hand, Hips, Torso, Backpack	1	3	2812 h
US-TMD	S, W, R, B, TR	A, M, G, GR, AL, P, AU, PX	Android	N/A	11	13	31 h
Collecty	W, R, BI, C, B, TR, TM, E	A, M, G, L	Android	N/A	N/A	15	242 h
NOR-TMD	B, M, TR, TM, BI, BO, C, E, I, O	A, M, M Unc., G, G Unc., RV, GV, RR, Q, GR, L, P, AR	Android, iOS	Hand, Pocket, Other	57	101	609 h

DL = device placement during data collection. N.UD = number of unique devices. N.UP = number of unique participants. Modes = (bus (B), tram (TM), train (TR), metro (M), car (C) Bike (BI), still (S), walk (W), run (R), motorcycle (MC), E-scooter (E), boat (BO) Inside (I), outside (O)). sensors = (accelerometer (a), magnetometer (M), gyroscope (G), pressure (P), gravity (GR), linear acceleration (L), ambient light (AL), orientation (O), rot. vector (RV), game rot. vector (GV), rotation rate (RR), quaternion (Q) Global positioning system (GPS), satellite reception (SR), Wi-Fi reception (WR) Mobile phone cell reception (CR), battery level (BL), battery temperature (BT), audio (AU), proximity (PX), built-in activity recognition (AR)).

diversity (three individuals) and is collected exclusively on one device type, significantly limiting its generalizability as diverse participants contribute data that reflect distinct travel behaviors and patterns, which can vary significantly based on demographics (Alharbi & Thornton, 2015), geographical regions (Nanchen et al., 2023), and individual habits (Weiss & Lockhart, 2012). In real-world scenarios, users interact with their devices in various ways, such as holding them in hand, placing them in a pocket, or storing them in a bag. These different placements significantly influence sensor readings (Reddy et al., 2010). Consequently, labeling the location of the device during data collection can provide valuable contextual information. Apart from the NOR-TMD dataset, the SHL dataset is the only other dataset that includes device location information during data collection. The SHL dataset provides four labels for device placement, whereas the NOR-TMD dataset offers three. Gjoreski et al. (2018) employed a rigorously controlled data collection protocol, mandating that devices be placed exclusively in specific, predefined locations. While this methodology is justifiable for minimizing noise and enhancing the reliability of comparisons, it can be argued that adopting a more flexible approach, as exemplified by the NOR-TMD dataset, may produce data that more accurately represents real-world usage scenarios. The NOR-TMD dataset incorporates an "other" label, allowing participants to place devices in unconventional locations such as backpacks, purses, or even the center console of a car while driving. This approach potentially leads to more diverse data, which may improve classification accuracy in real-world applications, where users are likely to store devices in a variety of locations. In contrast, solutions developed using the SHL dataset may struggle to accurately classify modes of transportation when devices are stored in locations outside those specifically included in their controlled study. The HTC dataset exhibits limited geographical diversity, as it was collected from only two distinct avenues. In contrast, the SHL dataset encompasses a broader geographical range, primarily spanning from Brighton, UK, to London, UK. The study by Erdelić et al. (2023) does not specify the devices used or the geographical locations from which the Collecty dataset was obtained. Similarly, Carpineti et al. (2018) provides no details regarding the geographical locations of its data collection. The NOR-TMD dataset includes data collected from two cities in Norway. While this is similar to the SHL dataset in terms of the number of locations, the geographical areas differ significantly. The SHL dataset covers regions in close proximity to each other, whereas the two cities in the NOR-TMD dataset, Oslo and Bodø, are much farther apart. Consequently, the NOR-TMD dataset can be considered more geographically diverse than the SHL dataset. US-TMD and Collecty address device diversity to some extent, incorporating data from 11 devices and spanning 13 and 15 participants, respectively. However, these datasets remain relatively small in scale compared to HTC and SHL, and their limited sensor diversity restricts their utility for exploring the full range of transport mode detection use cases. NOR-TMD addresses these gaps by providing data from 57 unique device models across iOS and Android platforms, collected from 101 diverse participants. This diversity enhances the dataset's generalizability (Gong et al., 2019), enabling it to support robust transport mode detection models capable of performing across a variety of hardware and software configurations. Including data from iOS devices is particularly significant, as none of the existing datasets incorporate iOS data, despite its prevalence in real-world usage (Grossi, 2019). The presence of iOS data enables the development of cross-platform solutions that remain performant regardless of the operating system, a critical requirement for practical deployment. In terms of transportation modes, existing datasets show notable limitations. For example, the HTC and SHL datasets cover major modes like buses and trains but omit others such as boats or e-scooters. Collecty offers more diversity by including less common modes like e-scooters, yet none of these datasets, including US-TMD, comprehensively address transport modes prevalent in specific geographical regions, such as seagoing vessels, which is problematic since this represents a regular mode of transportation in many areas Nikolic and Bierlaire (2017). NOR-TMD expands this scope by including 10 transport modes, such as boats and trams, alongside contextual labels like "inside" and "outside," enabling finer differentiation between activities. Sensor diversity is another critical area where NOR-TMD demonstrates an advantage. The SHL dataset provides extensive sensor data, including accelerometers, gyroscopes, and pressure sensors, but the HTC dataset is limited to three sensors, and Collecty includes only four. NOR-TMD balances sensor diversity with practical relevance, integrating data from 12 sensors on Android and 8 on iOS, including accelerometers, magnetometers, gyroscopes, and pressure sensors. This enables detailed analysis of sensor-specific contributions to transport mode detection performance while addressing gaps in other datasets.

Framework for feature evaluation and reduction

The proposed feature evaluation framework, EFR-TMD, introduces an ensemble-based methodology to identify and rank critical features for transport mode detection. EFR-TMD integrates multiple feature ranking techniques, including Permutation Importance (PI), Shapley Additive Explanations (SHAP), Mutual Information (MI), the ANOVA F-test, and Classification and Regression Tree (CART). These techniques individually assess the importance of features before their outputs are consolidated and normalized to produce a more robust and universally applicable feature importance score. Existing approaches to feature extraction and selection often rely on experimentation, domain expertise, or intuition (Chen et al., 2017; Nirmal et al., 2021; Sankaran et al., 2014). Some studies employ ablation analysis, where a model is trained on the full feature set, and features are systematically removed until a significant drop in accuracy is observed (Hemminki et al., 2013). Other studies employ sequential forward selection (SFS) which operates in the opposite manner, starting with an empty feature set and iteratively adding features until further additions fail to improve accuracy (Xiao et al., 2019). However, these methods are limited in their generalizability, as they typically indicate the relevance of features for a specific algorithm but do not necessarily translate well to other contexts. Some researchers have adopted more general feature importance techniques, such as InfoGainAttributeEval (Guvensan et al., 2017), but these approaches often produce significantly divergent results. Consequently, transferring insights to a different class of algorithm becomes challenging, necessitating time-consuming feature evaluation even when working with familiar datasets. To our knowledge, no existing work systematically ranks features in a manner that is both generic and widely applicable. This challenge is further compounded by the diverse range of sensors and aggregation functions utilized in the literature. EFR-TMD addresses these limitations by leveraging an ensemble of feature ranking techniques to deliver generic feature importance scores that are independent of specific models. As evidenced in our results, up to 75% of the least important features can be removed with only marginal impacts on accuracy (Tables 4 and 6). Moreover, the accuracy trends across different models when features are removed are remarkably consistent, underscoring the framework's ability to generalize its rankings beyond individual model architectures. This versatility makes EFR-TMD a valuable tool for transport mode detection applications.

Insights derived from EFR-TMD

Through the use of EFR-TMD we ranked the importance of sensors and aggregation functions individually both from Android and iOS platforms, for transport mode detection. The literature reveals considerable variation in sensor usage for classification. For instance, some studies combine accelerometers with magnetometers (Kamalian & Ferreira, 2022), while others exclude magnetometers in favor of linear acceleration and rotation vectors (Alotaibi, 2020). Our ranking of sensor importance on Android reveals that the least significant sensors are the game rotation vector, linear acceleration, and geomagnetic rotation vector, all averaging low ranks (Figure 7). Evaluation results (Table 3) demonstrate that excluding these sensors minimally impacts accuracy, and in some cases, improves it. Models trained with only the top 50% of ranked sensors maintain benchmark-level accuracy, indicating that sensors in the lower 50% contribute negligible new information. The accelerometer, pressure, and magnetic field sensors consistently rank highest. While many works utilize accelerometers (Cardoso et al., 2016; Nirmal et al., 2021), fewer include magnetic field (Chen et al., 2017) or pressure sensors (Sankaran et al., 2014). Incorporating these could significantly enhance accuracy, as suggested by our findings. Aggregation function importance rankings (Figure 8) indicate that range, interquartile range, and standard deviation are most impactful, with minimal variation among functions compared to sensors. Specific sensors, however, benefit more from certain functions. For example, variance ranks highly for pressure, while the first quartile is critical for gyroscope data. These insights highlight the potential benefits of tailoring aggregation functions to specific sensors. On iOS, results align closely with Android, with magnetic field, pressure, and accelerometer sensors ranking highest. Gravity, quaternion, and rotation rate sensors, however, are the least significant (Figure 11). Excluding the bottom 25% of sensors on iOS (e.g., gravity) minimally impacts accuracy (Table 5), but removing higher-ranked sensors, such as pressure or gyroscope, results in substantial accuracy drops. Our findings suggest that the rotation vector on Android, equivalent to a combination of rotation rate and quaternion on iOS, slightly outperforms the gyroscope in certain contexts. Aggregation function rankings on iOS (Figure 12) show similarities to Android, with interquartile range and minimum value performing well. However, differences arise due to platform-specific sensor behaviors and data characteristics. For instance, accelerometers on Android measure in meters per second squared (m/s^2), while iOS devices use gravitational increments. These discrepancies influence aggregation function effectiveness, as seen in the higher importance of quartile-based functions for iOS data. Our findings provide a ranked overview of sensor and aggregation function importance for transport mode detection across platforms. The accelerometer, magnetic field, and pressure sensors emerge as universally significant, with the gyroscope also proving beneficial. Reducing features improves training and inference efficiency, directly impacting energy consumption, a critical consideration for resource-constrained devices. As gyroscopes can consume significantly more power than other sensors (Yu et al., 2014), their inclusion should be carefully evaluated. By prioritizing high-importance features, researchers can develop lightweight, energy-efficient models without sacrificing accuracy.

Limitations

A significant limitation of NOR-TMD is the imbalance in data distribution across transport modes, devices, and users, which reflects the natural variability in participants' travel behaviors. While this contributes to the realism of the dataset, it also introduces potential biases that must be carefully managed during model development. Another challenge lies in the irregular sampling frequency of sensor data, a consequence of device-native collection frameworks. This variability can impact model reliability, necessitating preprocessing techniques such as interpolation to standardize input data. Although this issue can be mitigated through appropriate preprocessing, it remains a consideration in the dataset's design. Additionally, the dataset lacks location and audio data, which were collected but could not be included in the publicly available dataset due to privacy concerns. While the dataset spans a broader geographical area compared to existing datasets, incorporating data from a wider range of countries could further enhance its diversity in terms of physical environments, vehicle types, travel patterns, and user behaviors.

EFR-TMD also has its limitations. While the framework has been evaluated using the NOR-TMD dataset, its performance on datasets with differing characteristics has yet to be investigated. While EFR-TMD was tested across a wide array of commonly used algorithms, it does not encompass all possible models, raising the possibility that results may vary when applied to substantially different algorithms. Another limitation is the scope of features considered during the identification of universally applicable sensors and aggregation functions for transport mode detection. The evaluation included only features derived from the time domain. Had frequency-domain features been incorporated, the results might have differed. Another aspect is the sizes of the windows used for segregating the data. Furthermore, advanced processing techniques, such as smoothing, were not applied, as different sensors often require tailored processing methods. Our goal was to evaluate all features in a generic and consistent manner. However, the absence of sensor-specific preprocessing may have influenced the results. Introducing appropriate processing techniques for each sensor type could potentially alter the observed importance of individual sensors and provide further insights into their relative contributions.

Conclusion and future work

This study addresses key challenges in transport mode detection by introducing the novel NOR-TMD dataset and a systematic feature evaluation framework, EFR-TMD. The NOR-TMD dataset enhances the diversity and generalizability of transport mode detection research by incorporating data from both Android and iOS devices, expanding transportation mode coverage, and capturing real-world usage scenarios. Complementing this, the EFR-TMD framework employs an ensemble-based approach to identify and rank critical features, enabling the development of efficient and accurate transport mode detection models. Evaluation results reveal that up to 75% of the least important features identified by EFR-TMD can be removed with minimal impact on accuracy, significantly reducing computational costs and enabling resource-efficient, on-device processing. Despite limitations such as dataset imbalances and the focus on time-domain features, this work contributes valuable insights for the design of scalable, privacy-preserving, and generalizable solutions in intelligent transportation systems.

Future research should apply EFR-TMD to more diverse datasets and extend the analysis to include frequency-domain features, further broadening its applicability. Additional areas for future exploration include evaluating the applicability of EFR-TMD in other domains, such as human activity recognition or gesture recognition, where challenges related to feature selection and generalizability are similarly critical. Further studies should evaluate frequency-domain features to validate the generalizability of the identified sensors and aggregation functions. Expanding the analysis to include a wider range of algorithms would also provide deeper insights into the versatility of EFR-TMD. Additionally, while 10-second windows have been effective, exploring the relationship between window configurations and feature importance could reveal new optimization opportunities. Such investigations would further advance the robustness and applicability of the proposed framework. Future studies should also build on the insights derived from the feature evaluation and investigate the potential for cross-platform solutions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Alharbi, A. R., & Thornton, M. A. (2015). Demographic group classification of smart device users. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 481–486). IEEE.

Alotaibi, B. (2020). Transportation mode detection by embedded sensors based on ensemble learning. *IEEE Access*, *8*, 145552–145563. https://doi.org/10.1109/ACCESS.2020.3014901

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. *Bioinformatics (Oxford, England)*, *26*(10), 1340–1347. https://doi.org/10.1093/bioinformatics/btq134

Angel, S., Parent, J., Civco, D. L., Blei, A., & Potere, D. (2011). The dimensions of global urban expansion: Estimates and projections for all countries, 2000–2050. *75* (2), 53–107.

Asci, G., & Guvensan, M. A. (2019). A novel input set for LSTM-based transport mode detection. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (pp. 107–112). IEEE.

Asgari, F., & Clemencon, S. (2018). Transport mode detection when fine-grained and coarse-grained data meet. In *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)* (pp. 301–307). IEEE.

Aziz, A., Youkee, N. K., Ahmed, F. S., Tasnim, S., Fahim-Ul-Islam, M., & Chakrabarty, A. (2024). Enhancing transparency in transport mode detection: An interpretable ensemble model classifier. In *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)* (pp. 1–6). IEEE.

Bachir, D., Khodabandelou, G., Gauthier, V., El Yacoubi, M., Vachon, E (2019). Combining Bayesian inference and clustering for transport mode detection from sparse and noisy geolocation data. In U. Brefeld, et al. (Eds.), *Machine learning and knowledge discovery in databases* (pp. 569–584). Springer International Publishing.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, *5*(4), 537–550. https://doi.org/10.1109/72.298224

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Bulling, A., Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, *46*(3), 1–33. https://doi.org/10.1145/2499621

Burkhard, O., Becker, H., Weibel, R., & Axhausen, K. W. (2020). On the requirements on spatial accuracy and sampling rate for transport mode detection in view of a shift to passive signalling data. *Transportation Research Part C: Emerging Technologies*, *114*, 99–117. https://doi.org/10.1016/j.trc.2020.01.021

Cardoso, N., Madureira, J., & Pereira, N. (2016). *Smartphone-based transport mode detection for elderly care*. In 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom) (pp. 1–6). IEEE.

Carpineti, C., Lomonaco, V., Bedogni, L., Felice, M. D., & Bononi, L. (2018). *Custom dual transportation mode detection by smartphone devices exploiting sensor diversity*. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (pp. 367–372). IEEE.

Cavalcante, I. C., Meneguette, R. I., Torres, R. H., Mano, L. Y., Gonçalves, V. P., Ueyama, J., Pessin, G., Amvame Nze, G. D., & Rocha Filho, G. P. (2022). Federated system for transport mode detection. *Energies*, *15*(23), 9256. https://doi.org/10.3390/en15239256

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chen, K.-Y., Shah, R. C., Huang, J., & Nachman, L. (2017). Mago: Mode of transport inference using the hall-effect magnetic sensor and accelerometer. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(2), 1–23. 81–8:23. https://doi.org/10.1145/3090054

Cianciulli, D., Canfora, G., & Zimeo, E. (2017). Beacon-based context-aware architecture for crowd sensing public transportation scheduling and user habits. *Procedia Computer Science*, *109*, 1110–1115. https://doi.org/10.1016/j.procs.2017.05.451

Coskun, D., Incel, O. D., Ozgovde, A. (2015). Phone position/placement detection using accelerometer: Impact on activity recognition. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)* (pp. 1–6). IEEE.

Das, R. D., & Winter, S. (2018). A fuzzy logic based transport mode detection framework in urban environment. *Journal of Intelligent Transportation Systems*, *22*(6), 478–489. https://doi.org/10.1080/15472450.2018.1436968

Erdelić, M., Erdelić, T., & Carić, T. (2023). Dataset for multimodal transport analytics of smartphone users - Collecty. Data in *Brief*, *50*, 109481.

Ferreira, P., Zavgorodnii, C., & Veiga, L. (2020). edgeTrans - Edge transport mode detection. *Pervasive and Mobile Computing*, *69*, 101268. https://doi.org/10.1016/j.pmcj.2020.101268

Gjoreski, H., Ciliberto, M., Wang, L., Ordonez Morales, F. J., Mekki, S., Valentin, S., & Roggen, D. (2018). The University of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access*, *6*, 42592–42604. https://doi.org/10.1109/ACCESS.2018.2858933

Gong, Z., Zhong, P., & Hu, W. (2019). Diversity in machine learning. *IEEE Access*, *7*, 64323–64350. https://doi.org/10.1109/ACCESS.2019.2917620

Grossi, M. (2019). A sensor-centric survey on the development of smartphone measurement and sensing systems. *Measurement 135*, 572–592.

Guvensan, M. A., Dusun, B., Can, B., & Turkmen, H. I. (2017). A novel segment-based approach for improving classification performance of transport mode detection. *Sensors*, *18*(1), 87. https://doi.org/10.3390/s18010087

Hemminki, S., Nurmi, P., & Tarkoma, S. (2013). Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems* (pp. 1–14). Association for Computing Machinery.

Huang, Q., Zhang, J., Zeng, Z., He, D., Ye, X., & Chen, Y. (2023). PPDF-FedTMI: A federated learning-based transport mode inference model with privacy-preserving data fusion. *129*, 102845. https://doi.org/10.1016/j.simpat.2023.102845

Kamalian, M., & Ferreira, P. (2022). FogTMDetector - Fog based transport mode detection using smartphones. In *2022 IEEE 6th International Conference on Fog and Edge Computing (ICFEC)* (pp. 9–16). IEEE.

Kamalian, M., Ferreira, P., & Jul, E. (2022). A survey on local transport mode detection on the edge of the network. *Applied Intelligence*, *52*(14), 16021–16050. https://doi.org/10.1007/s10489-022-03214-y

Kaur, H., Pannu, H. S., & Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys*, *52*(4):1–36.

Kröger, J. L., Raschke, P., & Bhuiyan, T. R. (2019). Privacy implications of accelerometer data: A review of possible inferences. In *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy* (pp. 81–87). Association for Computing Machinery.

Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. T. (2010). A survey of mobile phone sensing. *48*(9), 140–150.

Liang, X., & Wang, G. (2017). A convolutional neural network for transportation mode detection based on smartphone platform. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)* (pp. 338–342). IEEE.

Lorintiu, O., & Vassilev, A. (n.d). Transportation mode recognition based on smartphone embedded sensors for carbon footprint estimation. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1976–1981). IEEE.

Lu, H., Pinaroc, M., Lv, M., Sun, S., Han, H., & Shah, R. C. (2019). Locomotion recognition using XGBoost and neural network ensemble. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers* (pp. 757–760). Association for Computing Machinery.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Mangiaracina, R., Perego, A., Salvadori, G., & Tumino, A. (2017). A comprehensive view of intelligent transport systems for urban smart mobility. *20*(1), 39–52. https://doi.org/10.1080/13675567.2016.1241220

Mastalerz, M. W., Malinowski, A., Kwiatkowski, S., Śniegula, A., & Wieczorek, B. (2020). Passenger BIBO detection with IoT support and machine learning techniques for intelligent transport systems. *Procedia Computer Science*, *176*, 3780–3793. https://doi.org/10.1016/j.procs.2020.09.009

Mathur, A., Zhang, T., Bhattacharya, S., Velickovic, P., Joffe, L., Lane, N. D., & Lio, P. (2018). Using deep data augmentation training to address software and hardware hetero-geneities in wearable and smartphone sensing devices. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 200–211). IEEE.

Matthes, P., & Springer, T. (2022). *Selecting resource-efficient ML models for transport mode detection on mobile devices*. In *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)* (pp. 135–141). IEEE.

Moreau, H., Vassilev, A., & Chen, L. (2022). The devil is in the details: An efficient convolutional neural network for transport mode detection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 12202–12212. https://doi.org/10.1109/TITS.2021.3110949

Muharemi, F., Syka, E., & Logofatu, D. (2020). Recognizing user's activity and transport mode detection: Maintaining low-power consumption. In P. Cellier & K. Driessens (Eds.), *Machine learning and knowledge discovery in databases* (pp. 21–37). Springer International Publishing.

Nanchen, A., Meegahapola, L., Droz, W., Gatica-Perez, D. (2023). Keep sensors in check: Disentangling country-level generalization issues in mobile sensor-based models with diversity scores. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 217–228). Association for Computing Machinery.

Nikolic, M., & Bierlaire, M. (2017). Review of transportation mode detection approaches based on smartphone data. In *17th Swiss Transport Research Conference*.

Nirmal, P., Disanayaka, I., Haputhanthri, D., & Wijayasiri, A. (2021). Transportation mode detection using crowd-sourced smartphone data. In *2021 28th Conference of Open Innovations Association (FRUCT)* (pp. 341–349). IEEE. https://doi.org/10.23919/FRUCT50888.2021.9347625

Oplenskedal, M., Taherkordi, A., Herrmann, P. (2021). ATARAXIS: A deep learning approach for hardwareless in-vehicle presence detection. In *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)* (pp. 1–10). IEEE.

Plötz, T., & Guan, Y. (2018). Deep learning for human activity recognition in mobile computing. *Computer Magazine*, *51*(5), 50–59. https://doi.org/10.1109/MC.2018.2381112

Prelipcean, A. C., Gidófalvi, G., & Susilo, Y. O. (2017). Transportation mode detection—An in-depth review of applicability and reliability. *Transport Reviews*, *37*(4), 442–464. https://doi.org/10.1080/01441647.2016.1246489

Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., Padma, V. (2020). Study the influence of normalization/transformation process on the accuracy of supervised classification. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 729–735). IEEE.

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2010). Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks*, *6*(2)13:, 1–27. https://doi.org/10.1145/1689239.1689243

Sadeghian, P., Håkansson, J., & Zhao, X. (2021). Review and evaluation of methods in transport mode detection based on GPS tracking data. *Journal of Traffic and Transportation Engineering (English Edition)*, *8*(4), 467–482. https://doi.org/10.1016/j.jtte.2021.04.004

Sadeghian, P., Zhao, X., Golshan, A., & Håkansson, J. (2022). A stepwise methodology for transport mode detection in GPS tracking data. *Travel Behaviour and Society*, *26*, 159–167. https://doi.org/10.1016/j.tbs.2021.10.004

Sankaran, K., Zhu, M., Guo, X. F., Ananda, A. L., Chan, M. C., Peh, L.-S. (2014). Using mobile phone barometer for low-power transportation context detection. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems* (pp. 191–205). Association for Computing Machinery.

Skretting, A., Grønli, T.-M. (2025). *NOR-TMD dataset for transport mode detection*. https://www.kaggle.com/datasets/scholarone/nor-tmd

Skretting, A., Grønli, T.-M., Majchrzak, T. A., Mateos, C., & Hirsch, M. (2024). *Enhanced transport mode recognition on mobile devices*.

Ståhle, L., & Wold, S. (1989). *Chemometrics and Intelligent Laboratory Systems*, *6*(4), 259–272. https://doi.org/10.1016/0169-7439(89)80095-4

Vu, T. H., Dung, L., & Wang, J.-C. (2016). Transportation mode detection on mobile devices using recurrent nets. In *Proceedings of the 24th ACM International Conference on Multimedia, New York, NY, USA* (pp. 392–396). Association for Computing Machinery.

Wahlström, J., Skog, I., & Händel, P. (2017). Smartphone-based vehicle telematics: A ten-year anniversary. *18*(10), 2802–2825.

Wang, B., Gao, L., & Juan, Z. (2018). Travel mode detection using GPS data and socioeconomic attributes based on a random forest classifier. *IEEE Transactions on Intelligent Transportation Systems*, *19*(5), 1547–1558. https://doi.org/10.1109/TITS.2017.2723523

Wang, J., Cao, B., Yu, P., Sun, L., Bao, W., & Zhu, X. (2018). Deep learning towards mobile applications. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)* (pp. 1385–1393). IEEE.

Weiss, G. M., & Lockhart, J. (2012). The impact of personalization on smartphone-based activity recognition. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Widhalm, P., Leodolter, M., & Brändle, N. (2018). Top in the lab, flop in the field? Evaluation of a sensor-based travel activity classifier with the SHL dataset. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, New York, NY, USA* (pp. 1479–1487). Association for Computing Machinery.

Widhalm, P., Nitsche, P., Brändie, N. (2012). Transport mode detection with realistic Smartphone sensor data. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (pp. 573–576). IEEE.

Xiao, G., Cheng, Q., & Zhang, C. (2019). Detecting travel modes using rule-based classification system and Gaussian process classifier. *IEEE Access*, *7*, 116741–116752. https://doi.org/10.1109/ACCESS.2019.2936443

Xiao, G., Juan, Z., & Zhang, C. (2015). Travel mode detection based on GPS track data and Bayesian networks. *54*, 14–22. https://doi.org/10.1016/j.compenvurbsys.2015.05.005

Yu, M.-C., Yu, T., Wang, S.-C., Lin, C.-J., & Chang, E. Y. (2014). Big data small footprint: The design of a low-power classifier for detecting transportation modes. *Proceedings of the VLDB Endowment*, *7*(13), 1429–1440.

Zannat, K. E., & Choudhury, C. F. (2019). Emerging big data sources for public transport planning: A systematic review on current state of art and future research directions. *Journal of the Indian Institute of Science*, *99*(4), 601–619. https://doi.org/10.1007/s41745-019-00125-9

Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. *12*(4), 1624–1639.

Zhang, X. Q. (2016). The trends, promises and challenges of urbanisation in the world. *Habitat International*, *54*, 241–252. https://doi.org/10.1016/j.habitatint.2015.11.018

Chapter 12

Article V

Implementation and Evaluation of Cross-Platform, Lightweight, On-Device Transport Mode Detection

Publication status: In review at Journal of IEEE Transactions on Intelligent Transportation Systems

Implementation and Evaluation of Cross-Platform, Lightweight, On-Device Transport Mode Detection

Anders Skretting, Tor-Morten Grønli, Raghava Rao Mukkamala

***Abstract*—In recent times, there has been a significant surge in the popularity of machine learning algorithms and their applications. Machine learning is applied to more and more aspects of daily life, extending their applications to numerous areas. One such application is public transportation. Lately, researchers have utilized smartphone sensor data to develop machine learning models for transport mode detection (TMD) in order to optimize public transportation in terms of streamlining ticket purchase, in addition to facilitating a more comprehensive understanding and analysis of human travel patterns. There are multiple works that propose solutions for TMD, however very few of these can be executed locally on device, let alone run on multiple operating systems. As such, in this paper we propose a lightweight cross-platform model for TMD, able to infer the mode of transportation on both Android and iOS. The model is implemented on-device and does not require any external equipment or a centralized solution to function. The proposed solution is then implemented on a selection of devices, which are tested on various public transport routes. We evaluate the actual accuracy based on predictions made on-device on board actual public transport vehicles, in addition to investigating the inference time and estimated energy consumption. The results of this work demonstrates the possibility of training lightweight platform-agnostic models for TMD using very few features. Moreover, the findings of this study can contribute towards improving existing in-vehicle presence detection solutions for automated ticketing.**

***Index Terms*—mobile, on-device, transport mode detection, real-life evaluation, sensors, machine learning, public transportation**

I. Introduction

TRANSPORT mode detection (TMD) is the capability to computationally infer a user's mode of transportation. That is, to classify which means of transportation a user is currently using. TMD is similar to human activity recognition (HAR) but while human activity recognition is concerned with inferring activities such as walking, jumping, running and falling. TMD focus on inferring contextual information regarding the mode of transportation a user is situated in, such as being on a bus, tram, train or similar. Accurate identification of contextual information such as the vehicle a user is located in, opens up for a variety of new applications in intelligent transportation such as enabling contextual marketing [1], facilitating improved travel pattern analysis [2] and automated fare collection [3]. TMD can act as a trigger in order to issue public transport tickets automatically by either initiating data transfer with a on-board device [4] or by correlating the users position to that of public transport vehicles, through a cloud service. Similarly, TMD can also facilitate enhanced travel metrics which can contribute significantly to travel pattern analysis. This kind of functionality will equip public transport operators, municipalities and other stakeholders with the necessary knowledge to streamline public transport scheduling and operational control, in addition to being able to plan and build more efficient routes, which in turn can lead to a better public transport offer for the general population.

Modern research into TMD relies to a large degree on machine learning algorithms trained on inertial mobile sensor data such as acceleration, magnetic field and gyroscopic data and a variety of solutions do already exists [5]–[7]. However, many of these solutions are designed to run in a centralized solution. This can be expensive for the operators in terms of hardware and maintenance. Furthermore, this approach can also be expensive for travelers relating to network costs. Moreover, centralized solutions can be problematic if the traveler resides in underground public transport systems which are often out of range of Wi-Fi networks [8]. An alternative approach is to run the TMD model locally, on the traveler's device, similar to that of the built-in activity recognition functionality in the Android and iOS operating systems. This way we avoid unnecessary data transfer, in addition to omit the need for large computational power from the operators side in order to receive and process the continuous stream of sensor data from travelers. Furthermore, on-device solutions alleviates privacy concerns [9] since the data collected on a device does not have to leave the device. Moreover, with the model running on-device we omit any kind of network latency and classifications can be made in near real-time [9]. Different on-device solutions have been proposed, however most of these solutions are evaluated by using a holdout set from their dataset used for training. While this is a sound approach if the data is representative to that of the real world, not many datasets are. In fact, there exists very few openly available datasets with sensor data collected onboard public transport vehicles with the purpose of training machine learning algorithms for TMD. Those that do exist [10]–[12] have various flaws, such as device homogeneity, lack of sensors, lack of transport modes, geographical limitations and so on. Therefore by evaluating models using a holdout set in this context can be misleading and the evaluated accuracy may not be true in a real-world setting. To our knowledge, all proposed solutions for on-device transport mode is also platform-specific, meaning that the models are tailored for working on either iOS or Android and not both. In our previous work we investigated the importance of all sensors on Android and iOS in the context of TMD and we saw that the most important sensors are the same across both platforms [13]. These results indicate that it should be possible to build a platform-agnostic model, able to run on both platforms. The contributions of this paper are the following:

- **Main objective:** Presenting a lightweight platform-agnostic model for local transport mode detection on resource constrained devices.
- **Second objective:** Evaluation of model performance on distinguishing between public transportation modes and alternative modes
- **Third objective:** Evaluation of model performance on differentiating between different public transportation modes

The model is trained on a large dataset consisting of data from over one hundred different iOS and Android devices. We developed and implemented the solution through native mobile applications on both Android and iOS and evaluated the actual accuracy, inference time, as well as estimating the energy impact. In the next section we review past and ongoing efforts towards TMD, with a focus on features and real-life implementations. In section III we describe the approach taken regarding feature selection, preprocessing, model architecture and implementation details. In section IV we present the results, while in section V we discuss our findings. In section VI we conclude our work with pointers towards future research.

II. Related Work

There have been substantial research efforts in the past years towards accurate TMD and a variety of approaches have been proposed. Some approaches takes advantage of coarse-grained network data such as call detail records (CDR) which generates coarse geographical location data [14]. Others take advantage of WAP (Wireless access protocol) [15] or GSM data and base station information [16]. However, while network information has proven useful, the most recent studies focus on location data (GPS) or inertial and ambient smartphone sensor data, either separately or in combination [15]. With the recent developments in the field of machine learning, it is no surprise that most works uses this data to train machine learning algorithms in order to classify the mode of transportation. There is a large variety in machine learning algorithms that have been applied in this field and the majority of the basic algorithms have been applied in the field of transport mode detection [17]. Some examples are decision trees [15], [18], random forest [19], k-nearest neighbours [20], bayesian inference [14], hidden markov model [5], support vector machines [17] and a variety of neural networks [21], [22]. In addition, more specialized algorithms such as gradient boosted trees [23], recurrent neural networks (RNN) [24] and convolutional neural networks (CNN) [22] have also been taken advantage of in order to achieve high accuracy transport mode inference. These models are then placed either on a server [23] which requires sensor data to be transmitted over the network in order to be classified, or on device [25].

These models require ambient and inertial sensor data in order to classify the mode of transportation and there are large discrepancies in terms of which sensors are being taken advantage of in the literature. A large portion of the proposed solutions employs acceleration in their models [15], [19], [22], [23], [25]. Some works combine acceleration with data from the magnetometer [19] or gyroscope [26], [27]. While other works also incorporates other sensors such as rotation vectors, orientation sensor, linear acceleration [1] or gravity [22]. Very few works incorporate air pressure [7], [21], [22] even though employing barometer data can result in higher accuracy [17]. TMD can be viewed as a sub-field of human activity recognition (HAR) [9] and data from these sensors are usually segmented using a window function [15], [28], [29], according to best practices for human activity recognition [30]. Window size is also linked to inference time [31] which is important to keep in mind when segmenting the data. This data is then aggregated using a variety of statistical functions such as variance [28], mean [15], [16], [19], [20], median [15], minimum and maximum values [15], [19], [20], kurtosis [15], standard deviation [19], percentiles [32], quartiles [29] and interquartile range [23]. Similar to the large diversity in feature engineering techniques, there are also infinite possibilities when it comes to tuning hyperparameters and the architecture of the models. A structured approach to find the optimal configuration is using grid search [33] which is a technique to train and evaluate a model for each combination of pre-defined parameters. However, very few works employ this approach due to the technique being very resource and time consuming.

While there have been a substantial research effort towards accurate TMD models based on smartphone sensor data, most of these models are trained on Android data [19], [23], [28], [32]. Some authors have also taken advantage of iOS data [7], [16], [21], however, a thorough literature search does not reveal any work that combines data from the two platforms in order to propose a cross-platform model for TMD able to run locally on-device. Building cross-platform solutions that relies on mobile sensor data is considered challenging [34] due to the considerable variation in hardware and software between the different models and manufacturers. The authors of Kos, Tomažič, and Umek [34] investigated the bias and noise in accelerometer and gyroscope readings of 116 different devices and found that while biases vary between different smartphone models and even within the same model, noise only varies between the different models but is stable within the same model. Their results exhibited larger differences between devices running iOS and Android than models running the same operating system, which can be a challenge when developing models intended to run on both operating systems. Mobile devices can be classified as resource-constrained systems due to their limited computational power and memory capacity [35] and aspects such as energy consumption needs to be taken into consideration. On-device energy consumption have been evaluated in real-life implementations of TMD models [18], [19]. In the work of [19], the authors concluded that it is possible to run machine learning models for TMD on-device, with an equivalent or less energy footprint than other common applications such as viewing the album application, receiving a phone call or using navigation services. However, these solutions employ GPS, which is known to consume significant amounts of energy [9]. In general, it is difficult to get an accurate measurement of the energy consumption on device since charging, discharge, temperature variations and battery load all affect the battery performance [36]. This difference in battery performance was also observed in [18]. Lastly, TMD can be seen in connection with in-vehicle presence detection and automated ticketing [37]. In [37] the authors show that a user's in-vehicle presence can be established using TMD over time in combination with GPS. However, other approaches for in-vehicle presence detection and automated ticketing approaches[38], [39] that utilizes Bluetooth could also be enhanced with accurate on-device TMD solutions.

III. Approach

This section presents details regarding data processing and architecture optimization. Additionally, the experiment design is explained in detail.

A. Feature Selection

This work is based on a previously collected dataset [7], [21]. This dataset consists of 577 million unique sensor events, collected by regular travelers using over 100 Android and iOS devices. Although, this dataset contains all available sensor data on the two platforms, only in a small subset of this dataset was interesting in order to make the model as lightweight as possible. Previous work showed that the most important sensors in terms of TMD were quite similar across both operating systems [13]. These results showed that the most important sensors on both Android and iOS were acceleration, pressure, magnetic field and gyroscope. This indicated the possibility to develop a cross-platform model, able to infer the mode of transportation on both platforms. As such, in order to develop a lightweight, cross-platform model for TMD only acceleration, magnetic field, pressure and gyroscope were used in this work. While these sensors all exists on both platform there are slight differences in the units of measurement used on the two platforms. For instance on Android, acceleration is measured in meters per second squared (m/s^2). On iOS on the other hand, the accelerometer provides data in g-forces and not directly in meters per second squared. Similarly, the pressure sensor on Android provides data in hectopascals (hPa) or millibars, while on iOS the data is provided in bars. As such, the data had to be converted to the same units of measurement.

The temporal data then had to be segmented into smaller chunks before it could be used for training our model. In previous work the importance of a variety of aggregation functions have been ranked [13]. Based on these results, only the seven highest ranked aggregation functions were used to aggregate the sensor data. As such, only the first, second and third quartile, in addition to the minimum value, range, standard deviation and the interquartile range were used for aggregating the data. Before segmenting the data the length and overlap of our window function had to be defined. In previous experiments 10-second windows with a 5-second overlap had been employed with decent results [7], [21], however we observed a significant increase in accuracy when we increased the window length, while simultaneously reducing the overlap. Window length directly impacts the inference time, resource consumption and model accuracy [40], Moreover, the pre-inference data collection period on-device have to mirror the size of the window so a larger window will increase the required time of on-device data collection. Through experimentation we found that the best trade-off between data collection time and accuracy was 15-second windows with one-second overlap, which is in line with the findings in [37], where they found an input length of 12.8 seconds to yield the best results. Aggregating each directional value, in addition to the magnitude, of the four selected sensors using the seven selected functions, our dataset used for training now consisted of only 91 different features. The class distribution of our training data was highly unbalanced and in order to balance our dataset before training, we took advantage of the Synthetic Minority Oversampling Technique (SMOTE) [41] in order to synthesize data in the minority classes.

B. Grid Search

After feature selection and preprocessing, we now had to find the optimal architecture of our model. An approach from the literature to evaluate different model configurations is grid search. Grid search is an approach where applicable configurations of hyperparameters are defined and then a model is trained for each possible combination of the predefined hyperparameters. Using this approach it is possible to test a large spectrum of architectures. That being said, there are potentially an infinite amount of possible configurations and it is therefore necessary to restrict the spectrum of hyperparameters. Moreover, since a new model is trained for each possible combination, grid search is highly resource and time consuming. Based on experimentation and previous work [33], we defined two sets of parameters related to different aspects of the model configuration, *optimization* and *architectural*. We then ran two separate grid searches based on the two sets of parameters. While others have incorporated a set number of epochs into their grid search [33], we instead established a threshold at 1000 epochs and then implemented an early stopping mechanism based on accuracy with a patience of five epochs, meaning that the training would finish after five epochs without any positive change in accuracy. Two separate grid searches enabled us to run the searches in parallel, which reduced the required time and resources significantly. One grid search was conducted for investigating the learning rate, optimizer and initializer and one search for finding the optimal numbers of nodes and layers. Note that for each value of nodes, we reduced the number by 10 % for each consecutive layer. This means that if the configuration was 800 nodes and two layers, the first layer would have 800, the second 720 and so on. Since we trained a cross-platform model on both Android and iOS data, we had to establish two holdout sets, one consisting of only Android data and one with only iOS data.

For each model trained during the grid search, we evaluated the model on both Android and iOS data. As such, for one combination of parameters, the accuracy based on the holdout sets would not necessarily be the same. Based on the grid search results we chose the architecture yielding the highest accuracy on both platforms by taking the intersection of conditions. This architecture yielded over 99% accuracy based on both the holdout set for Android and iOS. The high accuracy of the model suggested the possibility of overfitting which is when the model does not generalize well from observed data to unseen data [42]. This suspicion was confirmed during experimentation on real devices. As such, we implemented several regularization techniques. We implemented L_2 regularization in each hidden layer. The L_2 regularizer is controlled by the lambda (λ) parameter, which can take any value between 0 and infinity, where a larger value enforces more regularization. To mitigate the risk of overfitting, we employed an iterative approach to identify the maximum lambda value that did not lead to vanishing gradients. Through experimentation we found that a lambda value of 0.01 was suitable. We also included a dropout layer with a 50% dropout rate between each layer, as well as batch normalization. Dropout layers randomly set a fraction of input units to zero to prevent overfitting, while batch normalization normalizes and scales activations to enhance training efficiency and overall model performance. Using this approach we reached a more plausible accuracy of 83.83% and 82.66% for Android and iOS respectively. The model was designed to classify between *bus*, *metro*, *train*, *tram* and *alternative modes (ALTM)*. The ALTM class is used to differentiate between public transport vehicles and not and consists of a variety of data captured in different modes, such as walking, bicycling, e-scooter, ferry and car. Table I presents the different parameters and values explored, as well as the best reported parameter in each category, in addition to the best accuracy based on the two holdout sets.

TABLE I: Grid Search Hyperparameters

Optimization		
Parameter	**Values**	**Best**
Learning Rate	0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009	0.0004
Activation	relu	relu
Optimizer	adam, rmsprop	adam
Initializer	glorot_uniform, he_normal	glorot_uniform
Architectural		
Parameter	**Values**	**Best**
# Layers	1 to 15	14
# Nodes	100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000	800
Android accuracy		99.3 %
iOS accuracy		99.6 %

After defining our model architecture, we scaled the data in order to ensure that all features contribute equally to the model's performance. We selected the MinMax scaler for this task, which is one of the most widely recognized approaches for normalization and have outperformed other scaling techniques in previous works [43]. The MinMax scaler transforms the data by rescaling each feature to a range between 0 and 1. The MinMax scaler achieves this by subtracting the minimum value of the feature and then dividing by the range of the values. The MinMax scaler works as follows:

Given a feature x with minimum value x_{min}
and maximum value x_{max}, the MinMax scaling is defined as:

$$x_{\text{scaled}} = \frac{(x - x_{\text{min}})}{(x_{\text{max}} - x_{\text{min}})}$$

This transforms x to a new value x_{scaled} in the range $[0, 1]$.

This method preserves the relationships between the original data points while standardizing the scale, making it particularly suitable for models sensitive to feature magnitude differences, such as neural networks. Neural networks, such as the Multilayer Perceptron (MLP) used in this work, uses gradient decent-based methods for training. Unscaled features can cause gradients to be disproportionately large or small and by scaling we eliminate this problem.

C. Experiment Design

To test the model in a real-world setting, two native applications were developed, one for Android and one for iOS. These applications functioned as wrappers for our cross-platform model and enabled measurements of inference time and energy consumption. In addition, through the interface it was possible to label wrongly classified modes. For the solution to be practical, it was important to investigate whether the actual, observed accuracy of the implemented model deviated from the statistical accuracy derived from the holdout set of the training data. Furthermore, investigations into performance aspects such as the inference time on-device and the energy consumption were relevant as well. Regardless of accuracy, if the time to make a classification is excessively prolonged or if the energy usage is disproportionately high, it would be impractical to implement in a real-world scenario. With this type of real-world experiment, where numerous factors can impact the results, the goal was to evaluate the applicability of the solution across a variety of devices. One challenge related to device heterogeneity is that software and hardware can vary significantly across different manufacturers and models, which in turn can prove challenging when building cross-platform solutions, especially when it comes to machine learning, where minute differences might tip the scale over in a different outcome. As such, a small but diverse set of devices were used during experimentation. This would enable the detection of any potential malfunction of the application, as well as ensuring that the adjustments made to wrong classifications were exact. In order to assess our cross-platform model in a real-world context we installed the applications on a diverse set of devices, representing both older and newer devices of both platforms. Details about the devices used for this experiment can be viewed in Table II. These devices were then used to evaluate aspects relating to accuracy, inference time and energy consumption, in all the aforementioned modes of transportation.

TABLE II: Device Overview

Device	OS	Version	Battery Capacity (mAh)
Sony Xperia 1 (J9110)	Android	11 (30)	3330
Pixel 7a	Android	14 (34)	4385
Samsung Galaxy S21 FE (SM-G990B)	Android	14 (34)	4500
Samsung Galaxy S22 Ultra (SM-S908B)	Android	14 (34)	5000
Samsung Galaxy S23 (SM-S911B)	Android	14 (34)	3900
iPhone 8	iOS	16.7.10	1821
iPhone 13	iOS	17.2.1	3240

Other than software and hardware challenges, there are also challenges related to positioning. Some of the features the model is trained on are directional sensor values. This means that device movement in specific direction during inference may influence the results. Furthermore, public transport vehicles, especially trains, trams and subways may have seats facing different directions, such as forwards, backwards and sideways, which also could impact the results. Moreover, the actual position of the device can also vary, whether it is held in hand or stored in a pocket or backpack. While the training dataset was labeled with device position (hand, pocket, other), the seating direction was not labeled. In a production environment, this kind of solution would have to work regardless of device direction or placement, as such we used data labeled with both hand, pocket and other. On the other hand, it is difficult to systematically evaluate all possible positions of a device in a real-life context. Therefore, for this experiment we evaluated the model while allowing all natural directions, however always with the device in hand.

IV. Results

The real-life test scenario resulted in a total of 4999 classifications. In this section we present results in terms of actual accuracy, in addition to measured inference time and energy usage. We separate the results from Android and iOS. While it is the exact same model tested it is relevant to investigate the differences between the two platforms.

A. Model Accuracy

As previously mentioned, after each classification a file was stored on the device holding the classification from the model, in addition to the eventual correction made by the user. Based on this information it is possible to construct a classification report and a confusion matrix for both platforms.

1) Android: Figure 1 displays the confusion matrix derived from the holdout set with data from Android devices, while figure 2 shows the confusion matrix based on the real-world experiment. Table III exhibits the classification reports of the model based on the holdout set and based on the real-life implementation. From the classification report we can see that the model is able to successfully classify bus, metro and alternative modes with high accuracy. Classification of train and tram is suboptimal. A highly encouraging observation is that there are very few wrong classifications between the public transportation modes and other. If we group all the public transport classes together and look at the classification accuracy between public transport and alternative we achieve roughly 99% accuracy.

Fig. 1: Confusion Matrix Android - Holdout set of original data

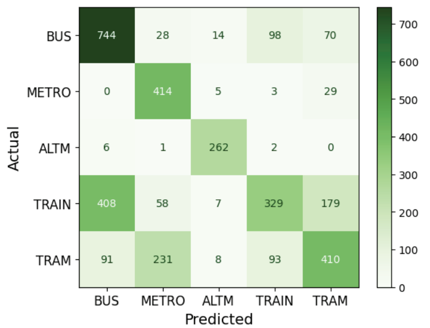


Fig. 2: Confusion Matrix Android - Real-life experiment

2) iOS: Figure 3 and figure 4 shows the confusion matrix derived from the holdout set with data from iOS devices and from the real-world experiment, respectively. Table IV shows the classification report based on both the holdout set, as well as the real-world experiment. Analogous to the Android results, a high performance can be observed when classifying bus, metro and other, while the modes train and tram is more difficult to classify. Similar to that of the results pertaining to data collected on Android devices, by grouping the public transportation modes together, we are able

Class	Precision	Recall	F1-Score	Support
Holdout set				
BUS	0.92	0.86	0.89	97846
METRO	0.95	0.82	0.88	51346
ALTM	0.93	0.86	0.89	59951
TRAIN	0.35	0.89	0.51	5932
TRAM	0.48	0.76	0.59	13300
Accuracy			0.85	228375
Real-life Experiment				
BUS	0.60	0.78	0.68	954
METRO	0.57	0.92	0.70	451
ALTM	0.89	0.97	0.92	271
TRAIN	0.63	0.34	0.44	981
TRAM	0.60	0.49	0.54	833
Accuracy			0.62	3490

TABLE III: Classification Report Android

Class	Precision	Recall	F1-Score	Support
Holdout set				
BUS	0.97	0.82	0.89	116187
METRO	0.82	0.85	0.84	24784
ALTM	0.86	0.86	0.86	40960
TRAIN	0.53	0.86	0.66	14742
TRAM	0.47	0.79	0.59	11358
Accuracy			0.83	208031
Real-life Experiment				
BUS	0.93	0.89	0.90	1036
METRO	0.60	0.86	0.71	306
ALTM	1.00	0.84	0.91	176
TRAIN	0.60	0.11	0.18	509
TRAM	0.37	0.65	0.47	482
Accuracy			0.67	2509

TABLE IV: Classification Report iOS

to accurately distinguish between public transport vehicles and the ALTM class on iOS devices as well.

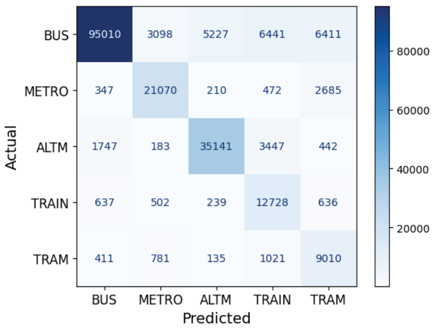


Fig. 3: Confusion Matrix iOS - Holdout set of original data

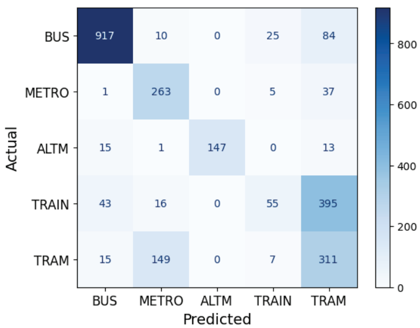


Fig. 4: Confusion Matrix iOS - Real-life test

3) Correlation of Results and Training Data: We observed a suboptimal performance when classifying tram and train on both platforms and a likely cause for this is a lack of training data in these modes. Figure 5 shows the correlation between the amount of training data and the number of incorrect classifications across both platforms. We can observe that the modes tram and train have significantly fewer training samples than the other modes. However, although there are considerable more data captured onboard busses than metro and other, there are also more incorrect classifications than metro and ALTM. As such, the lack of data is probably not the only cause for incorrect classifications and it could be that the relationship between sensor values and the modes train and tram, is less explicit than it is for instance on metro.

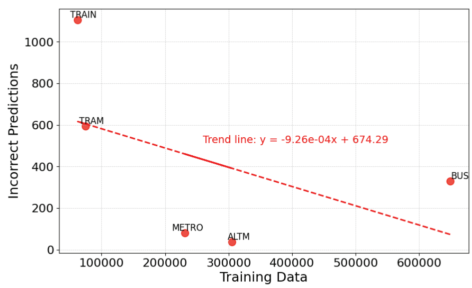


Fig. 5: Correlation of Results and Training Data - Both Platforms

B. Inference Time and Energy Consumption

The duration for the model to make each classification was also recorded. The average inference time for the model to make a classification on Android devices was 5.31 milliseconds (ms). However, the inference time varied significantly between less than 1 ms to 74 ms. For iOS devices the average inference time was 2.05 ms and ranged from less than 1 ms to 68 ms. Figure 6 shows the distribution of the registered inference times across all devices. We can see that there are significant differences in the time the model need to make a classification between the different devices. However, in terms of overall viability, less than 74 ms is well within acceptable range. Figure 7 shows the relationships between the inference times and the predicted mode. It is interesting to note that there are fairly large variations in the inference time when predicting different modes. We also estimated the energy consumption based on three of the Android devices. While energy consumption is not the main focus of this work, it is important to gauge the consumed energy in order to make an informed decision about the viability of this kind of solution. Using a hardware-based energy measurement approach was difficult as we wanted to assess the actual consumption during inference onboard public transport vehicles. We observed a relationship between the inference times and the different modes of transportation. As such, the energy consumption could also be dependent on the mode and a stationary energy estimation using hardware-based measurement would not necessarily be accurate. We used the Android Debug Bridge (ADB) to extract energy measurements from our application and estimated a total energy consumption for our application running the model to be between 0.83 % to 2.79% per hour. This is a rough estimate, including energy consumed by the screen, which was on continuously during experimentation in order to correctly label the results. The energy consumption of the screen is significant and makes up most of the energy consumed. For some devices, such as the SM-S911B The energy consumed by the screen during our experiment is registered to be 143 mAh per hour for our application, while the total energy consumed is reported to be 177 mAh per hour.

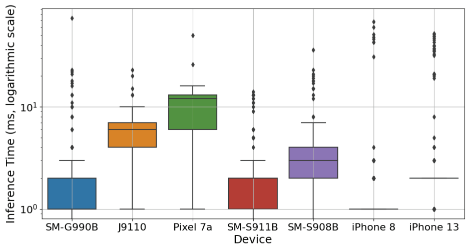


Fig. 6: Inference Times Across All Devices

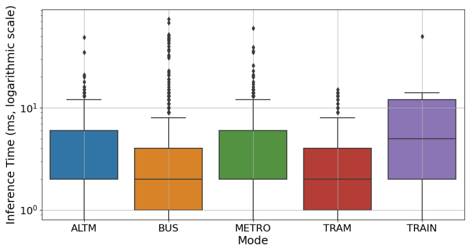


Fig. 7: Inference Times Across All Modes

Figure 8 shows the relationship between the Android devices and correct and incorrect classifications in the different modes. It is interesting to note that while most devices are able to classify the mode bus with high accuracy, the amount of incorrect classifications in this mode with the Pixel 7a is significantly higher. In relation to the inference time and energy consumption we also saw that the inference time for the Pixel 7a was significantly longer, while at the same time having a much smaller energy footprint. We did not go deeper into the details of this, however it could be aspects relating to the hardware or software of the Pixel that influence this.

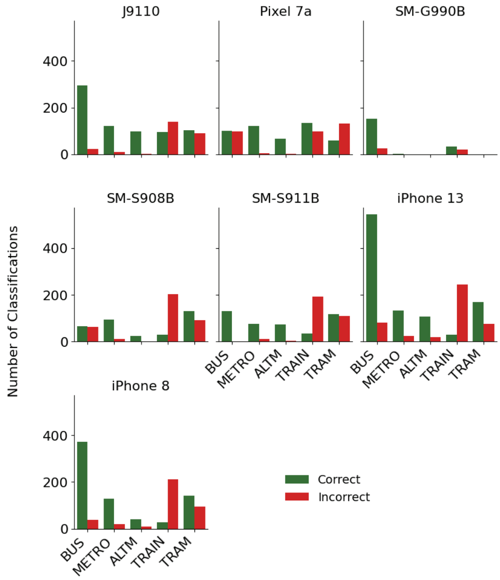


Fig. 8: Correct and Incorrect Classifications per Device

V. Discussion

In this work we presented, to our knowledge, the first neural networok-based platform-agnostic model for local TMD, able to infer the mode of transportation on-device on both Android and iOS. We evaluated the model in a real-life context and investigated aspects such as actual accuracy and inference time. In addition, we gauged the energy consumption of the Android devices. We saw that the model is able to classify modes such as bus, metro and other with high accuracy, however the modes train and tram lead to suboptimal results. Although, most other works only evaluate their solutions based on cross-validation [15], [23] or holdout sets [22], very few works implement their solutions in a real-life setting. From the works that have been implemented and tested in a real-life setting [18], [19], we see that discrepancies between evaluations based on the original data and real-life data are present in some of these works as well [18]. This is likely a result of overfitting, similar to what we observed for our model. While our model struggled with discerning train and tram, we see that the random forest classifier used in a real-world setting struggled with the mode bus [18]. As the authors of that work points out, running models on real transport vehicles, using real devices is much more challenging and the results are rarely as good as when evaluated in lab setting or simulated environment. This is our experience as well and as we have seen in this work and previous work [18], evaluations based on simulations can be misleading. While overfitting was a problem, another likely reason for our model to struggle with the modes train and tram is lack of training data within these modes. We saw that the model was trained on significantly less data captured onboard trams and trains than for the other modes. We did balance our dataset using SMOTE [41], so that there were an equal number of samples of all classes, however, for modes with a smaller amount of original samples, there would then be a much larger amount of synthetic data, compared to classes with more original data. By synthesizing data we do not expand the spectrum of possible values, instead we generate more similar values. As such, more data collected within the minority modes would likely increase the overall accuracy. It is also important to keep in mind that the goal of this work was to investigate whether a lightweight, cross-platform model for on-device TMD could be achieved. As such, we deliberately reduced the available features significantly to only employing four sensors and seven aggregation functions. By incorporating more features, such as rotation vectors or the amplitude of ambient noise [7] we might have achieved a higher accuracy in the problematic classes.

Although, classification on trains and trams were suboptimal, looking at this in the context of automated ticking, we achieved a very high accuracy on both platforms when classifying whether a traveler was on a public transportation vehicle or not. Previous work have attempted to achieve automated ticketing using Bluetooth beacons and similar [38], [39], however the main issue with these approaches is that the range of the Bluetooth signal transcends the confines of the vehicle. This leads to people walking next to the vehicle to be in range of the Bluetooth signal and as such subject to ticketing. The results from our work could be taken advantage of in combination with a Bluetooth approach to significantly limit this problem. On-device TMD can also be used in combination with GPS to achieve in-vehicle presence detection [37]. Our model also performed well in terms of inference time, we saw that the average inference time was between 2 and 6 ms and that the longest recorded inference time was 74 ms, which is well within acceptable bounds compared to previous work [31], [37]. TMD takes to a large degree advantage of similar techniques and approaches as activity recognition. Although different algorithm and different devices, previous work on TMD have reported average inference times of 32 ms to 59 ms [37], while work on activity recognition algorithms running on device have reported inference times between 228 ms and 292ms [31]. We saw that the average inference time varied significantly from device to device. The pre-inference data collection being conducted on device is to a large degree dependent on the operating system allocating resources to the sensor service and as such the amount of data amassed in the collection period varies significantly. However, this should not influence the inference time since the time is measured *after* the data is collected and processed and as such the amount of data fed into the model is always the same. All the devices have different hardware and software, and a discrepancy in inference time between the different devices is to be expected. However, an interesting observation that we made was that the inference time also varied depending on the mode of transportation being classified. This is harder to explain and we did not investigate this in-depth, although a possible explanation is that the modes requiring longer time might be more difficult for the model to classify.

We also gauged the energy consumption using three different Android devices, similar to what was done in [18], [19]. We see that the energy usage of our proposed solution is similar to the reported energy consumption in [18], [19]. However, our measurements include energy consumed from the screen, which was on continuously, while in [18], the screen was off. Thus, our estimates promises a more energy efficient solution than that of [18]. That being said, there are of course nuances to this. First of all, the device would have to run the operating system and everything else on the device for the application to function which would require energy. Moreover, the battery's behavior changes over time and is subject to temperature differences, in addition to behave differently depending on the general load [36]. As such, this is not an accurate measure but more of a rough estimate. An interesting aspect of the energy consumption in relation to the inference time is that the device with the fastest inference time (SM-S911B) was also the device with the highest reported energy expenditure. Similarly the device with the slowest inference time (Pixel 7a) had the lowest energy usage. This may be a result of hardware and software differences between the devices. These estimates indicates that the energy consumption on device is fairly low, which attests to the viability of this kind of solution running on device.

VI. Conclusion and Future Work

In this paper we presented to the best of our knowledge the first local platform-agnostic neural network for TMD. The lightweight model is able to run solely on-device, without the need of any centralized solution or network transmission. The model performed well when distinguishing between the classes bus, metro and alternative modes and if all the public transport classes are grouped together we are able to distinguish between alternative modes and public transportation modes with an accuracy of 99% on both Android and iOS. This can be taken advantage of in order to improve existing solutions for automated ticketing, such as Bluetooth- or GPS-based solutions. We evaluated the inference time, which was between less than 1 ms up to 74 ms across all devices and both platforms. The energy consumed by our implementation on Android, including both pre-inference sensor data collection, pre-processing of the collected data and the inference itself was negligible, with the highest recorded energy consumption of 34 mAh per hour. The results of this work shows that it is possible to provide TMD, either stand-alone or as a component to improve automated ticketing with a minimal energy footprint on-device without the need of external equipment or a centralized solution.

In the future we would like to improve our model by amassing more data captured within the problematic modes, as well as conducting a deeper investigation into the problem of overfitting. Furthermore, it could be worthwhile to consider applying reinforcement learning techniques in order to calibrate the model to a given traveler or device. Future investigations should also incorporate a more in-depth analysis of energy consumption on both platforms. Moreover, it would be interesting to investigate differences in energy consumption for different machine learning algorithms running on device. As mentioned, an applied implication is that TMD could potentially be taken advantage of in order to improve existing works on automated ticketing and in-vehicle presence detection and it could be insightful to explore this combination in a real-life setting.

References

[1] B. Alotaibi, "Transportation Mode Detection by Embedded Sensors Based on Ensemble Learning," *IEEE Access*, vol. 8, pp. 145 552–145 563, 2020.

[2] Z. Wang, S. Y. He, and Y. Leung, "Applying mobile phone data to travel behaviour research: A literature review," *Travel Behaviour and Society*, vol. 11, pp. 141–155, Apr. 2018.

[3] M. Bieler, A. Skretting, P. Büdinger, and T.-M. Grønli, "Survey of Automated Fare Collection Solutions in Public Transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 248–14 266, Sep. 2022.

[4] M. Oplenskedal, P. Herrmann, and A. Taherkordi, "DeepMatch2: A comprehensive deep learning-based approach for in-vehicle presence detection," en, *Information Systems*, vol. 108, p. 101 927, Sep. 2022.

[5] P. Widhalm, P. Nitsche, and N. Brändie, "Transport mode detection with realistic Smartphone sensor data," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov. 2012, pp. 573–576.

[6] A. Efthymiou, E. N. Barmpounakis, D. Efthymiou, and E. I. Vlahogianni, "Transportation Mode Detection from Low-Power Smartphone Sensors Using Tree-Based Ensembles," en, *Journal of Big Data Analytics in Transportation*, vol. 1, no. 1, pp. 57–69, Jun. 2019.

[7] A. Skretting, T.-M. Grønli, T. A. Majchrzak, C. Mateos, and M. Hirsch, *Enhanced Transport Mode Recognition on Mobile Devices*, eng. Jan. 2024.

[8] T. Stockx, B. Hecht, and J. Schöning, "SubwayPS: Towards smartphone positioning in underground public transportation systems," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '14, New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 93–102.

[9] M. Kamalian, P. Ferreira, and E. Jul, "A survey on local transport mode detection on the edge of the network," en, *Applied Intelligence*, vol. 52, no. 14, pp. 16 021–16 050, Nov. 2022.

[10] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, "Big data small footprint: The design of a low-power classifier for detecting transportation modes," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, Aug. 2014.

[11] H. Gjoreski, M. Ciliberto, L. Wang, *et al.*, "The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics With Mobile Devices," *IEEE Access*, vol. 6, pp. 42 592–42 604, 2018.

[12] C. Carpineti, V. Lomonaco, L. Bedogni, M. D. Felice, and L. Bononi, "Custom Dual Transportation Mode Detection By Smartphone Devices Exploiting Sensor Diversity," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2018, pp. 367–372.

[13] A. Skretting, T.-M. Grønli, and R. R. Mukkamala, "Feature importance from novel dataset for on-device transport mode detection," *Journal of Intelligent Transportation Systems*, Dec. 2023, Manuscript under review.

[14] D. Bachir, G. Khodabandelou, V. Gauthier, M. El Yacoubi, and E. Vachon, "Combining Bayesian Inference and Clustering for Transport Mode Detection from Sparse and Noisy Geolocation Data," en, in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Curry, E. Daly, *et al.*, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 569–584.

[15] N. Cardoso, J. Madureira, and N. Pereira, "Smartphone-based transport mode detection for elderly care," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Sep. 2016, pp. 1–6.

[16] F. Asgari and S. Clemencon, "Transport Mode Detection when Fine-grained and Coarse-grained Data Meet," in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Sep. 2018, pp. 301–307.

[17] M. Nikolic and M. Bierlaire, "Review of transportation mode detection approaches based on smartphone data," in *17th Swiss Transport Research Conference*, 2017.

[18] P. Ferreira, C. Zavgorodnii, and L. Veiga, "edgeTrans - Edge transport mode detection," *Pervasive and Mobile Computing*, vol. 69, p. 101 268, Nov. 2020.

[19] M. Kamalian and P. Ferreira, "FogTMDetector - Fog Based Transport Mode Detection using Smartphones," in *2022 IEEE 6th International Conference on Fog and Edge Computing (ICFEC)*, May 2022, pp. 9–16.

[20] P. Sadeghian, X. Zhao, A. Golshan, and J. Håkansson, "A stepwise methodology for transport mode detection in GPS tracking data," en, *Travel Behaviour and Society*, vol. 26, pp. 159–167, Jan. 2022.

[21] A. Skretting and T.-M. Grønli, "Neural Network for Public Transport Mode Inference on Mobile Devices," en, in *Mobile Web and Intelligent Information Systems*, I. Awan, M. Younas, and A. Poniszewska-Marańda, Eds., Cham: Springer International Publishing, 2022, pp. 65–78.

[22] H. Moreau, A. Vassilev, and L. Chen, "The Devil is in the Details: An Efficient Convolutional Neural Network for Transport Mode Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 202–12 212, Aug. 2022.

[23] P. Nirmal, I. Disanayaka, D. Haputhanthri, and A. Wijayasiri, "Transportation Mode Detection Using Crowdsourced Smartphone Data," in *2021 28th Conference of Open Innovations Association (FRUCT)*, Jan. 2021, pp. 341–349.

[24] T. H. Vu, L. Dung, and J.-C. Wang, "Transportation Mode Detection on Mobile Devices Using Recurrent Nets," in *Proceedings of the 24th ACM international conference on Multimedia*, ser. MM '16, New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 392–396.

[25] I. C. Cavalcante, R. I. Meneguette, R. H. Torres, *et al.*, "Federated System for Transport Mode Detection," en, *Energies*, vol. 15, no. 23, p. 9256, Jan. 2022.

[26] M. A. Guvensan, B. Dusun, B. Can, and H. I. Turkmen, "A Novel Segment-Based Approach for Improving Classification Performance of Transport Mode Detection," en, *Sensors*, vol. 18, no. 1, p. 87, Jan. 2018.

[27] G. Asci and M. A. Guvensan, "A Novel Input Set for LSTM-Based Transport Mode Detection," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2019, pp. 107–112.

[28] P. Widhalm, M. Leodolter, and N. Brändle, "Top in the Lab, Flop in the Field? Evaluation of a Sensor-based Travel Activity Classifier with the SHL Dataset," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp '18, New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 1479–1487.

[29] O. Burkhard, H. Becker, R. Weibel, and K. W. Axhausen, "On the requirements on spatial accuracy and sampling rate for transport mode detection in view of a shift to passive signalling data," en, *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 99–117, May 2020.

[30] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys*, vol. 46, no. 3, 33:1–33:33, Jan. 2014.

[31] X. Cheng, L. Zhang, Y. Tang, Y. Liu, H. Wu, and J. He, "Real-Time Human Activity Recognition Using Conditionally Parametrized Convolutions on Mobile and Wearable Devices," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5889–5901, Mar. 2022.

[32] R. D. Das and S. Winter, "A fuzzy logic based transport mode detection framework in urban environment," *Journal of Intelligent Transportation Systems*, vol. 22, no. 6, pp. 478–489, Nov. 2018.

[33] X. Jiang and C. Xu, "Deep Learning and Machine Learning with Grid Search to Predict Later Occurrence of Breast Cancer Metastasis Using Clinical Data," *Journal of Clinical Medicine*, vol. 11, no. 19, p. 5772, Sep. 2022.

[34] A. Kos, S. Tomažič, and A. Umek, "Evaluation of Smartphone Inertial Sensor Performance for Cross-Platform Mobile Applications," en, *Sensors*, vol. 16, no. 4, p. 477, Apr. 2016.

[35] T. S. Ajani, A. L. Imoize, and A. A. Atayero, "An Overview of Machine Learning within Embedded and Mobile Devices–Optimizations and Applications," en, *Sensors*, vol. 21, no. 13, p. 4412, Jan. 2021.

[36] A. McIntosh, S. Hassan, and A. Hindle, "What can Android mobile app developers do about the energy consumption of machine learning?" en, *Empirical Software Engineering*, vol. 24, no. 2, pp. 562–601, Apr. 2019.

[37] M. Oplenskedal, A. Taherkordi, and P. Herrmann, "ATARAXIS: A Deep Learning Approach for Hardwareless In-Vehicle Presence Detection," in *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*, Dec. 2021, pp. 01–10.

[38] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselböck, and N. Höfler, "Bluetooth Low Energy as enabling technology for Be-In/Be-Out systems," in *13th IEEE CCNC*, Jan. 2016, pp. 423–428.

[39] B. Wieczorek and A. Poniszewska-Marańda, "Be in/Be out model for intelligent transport in SmartCity approach," in *Proc. 17th MoMM2019*, Munich Germany: ACM, Dec. 2, 2019, pp. 226–230.

[40] P. Matthes and T. Springer, "Selecting Resource-Efficient ML Models for Transport Mode Detection on Mobile Devices," in *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, Nov. 2022, pp. 135–141.

[41] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Oversampling Technique," *JAIR*, vol. 16, pp. 321–357, Jun. 1, 2002. arXiv: 1106.1813 [cs].

[42] X. Ying, "An Overview of Overfitting and its Solutions," en, *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022 022, Feb. 2019.

[43] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Aug. 2020, pp. 729–735.

Anders Skretting received his M.Sc in Applied Computer Science from Kristiania University College, Norway and is currently pursuing his PhD in Applied Information Technology at Kristiania University College, Norway. His primary research interests are applied machine learning in combination with mobile computing.

Tor-Morten Grønli received his PhD degree in computer science from Brunel University. He is currently a Professor at the Department of Technology, Kristiania University College, Norway, and founding director of the Research Group for Applied Computer Science and the Mobile Technology Lab at the Institute of Technology. His primary research interests are mobile computing, internet of things, and software architectures.

Raghava Rao Mukkamala is the director of the Centre for Business Data Analytics and an associate professor at the Department of Digitalization, Copenhagen Business School. He is also the Study Line Coordinator for the new Masters Program in Data Science. Raghava's current research focus is on the interdisciplinary approach to big data analytics. Combining formal/mathematical modeling approaches with data/text mining techniques and machine learning methodologies. Raghava holds a Ph.D. degree in Computer Science and an M.Sc degree in Information Technology, both from IT University of Copenhagen, Denmark and a Bachelor of Technology degree from Jawaharlal Nehru Technological University, India.

Kristiania University of Applied Sciences
PO Box 1190 Sentrum
NO-0107 Oslo

PhD Dissertations Kristiania 2025:3
ISSN: 3084-1488 (print)
ISSN: 3084-147X (online)
ISBN: 978-82-93953-11-1 (print)
ISBN: 978-82-93953-12-8 (online)