Mining User Transport Behavior from Smartphones

Servizi, Valentino

Publication date:
2021

Document Version
Publisher's PDF, also known as Version of record

DTU Management
Department of Technology, Management and Economics

Mining User Transport Behavior from Smartphones

PhD Thesis

Valentino Servizi

December 2021

DTU Management
Department of Technology, Management and Economics

Mining User Transport Behavior from Smartphones

PhD Thesis
December 2021

By
Valentino Servizi

Main supervisor:	Francisco Camara Pereira, Professor at Department of Technology, Management and Economics, Technical University of Denmark.
Co-supervisor:	Otto Anker Nielsen, Professor at Department of Technology, Management and Economics, Technical University of Denmark.
University:	Technical University of Denmark
Department:	DTU Management, Department of Technology, Management and Economics
Division:	Transport Division
Group:	Machine Learning for Smart Mobility Group

Cover photo: Modern cityscape diorama and various transportation network, conceptual abstract image / stock.adobe.com, standard license file #116604161, date 18.10.2018 13.35, unlimited web, social-media, email, mobile; 500,000 prints.

Published by: DTU, Department of Technology, Management and Economics, Akademivej, Building 358, DK-2800 Kongens Lyngby, Denmark.
www.man.dtu.dk

Preface

This PhD thesis entitled *Mining User Transport Behavior from Smartphones* is submitted to meet the requirements for obtaining a PhD degree at the Department of Technology, Management and Economics, DTU Management, Technical University of Denmark. The PhD project was supervised by Professor Francisco Camara Pereira and co-supervised by Professor Otto Anker Nielsen, both from DTU Management. The thesis is paper-based and consists of the chapters listed in the tables of content, including separate chapters for each of the following papers:

Paper A: V. Servizi, C. F. Pereira, K. M. Anderson, and A. O. Nielsen (2021). "Transport behavior-mining from smartphones: a review." In: *European Transport Research Review*. DOI: 10.1186/s12544-021-00516-z. URL: https://doi.org/10.1186/s12544-021-00516-z.

Paper B: V. Servizi, N. C. Petersen, F. C. Pereira, and O. A. Nielsen (2020). "Stop detection for smartphone-based travel surveys using geo-spatial context and artificial neural networks". In: *Transportation Research Part C: Emerging Technologies* 121, p. 102834. DOI: 10.1016/j.trc.2020.102834. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X20307385.

Paper C: V. Servizi, D. R. Persson, F. C. Pereira, H. Villadsen, P. Bækgaard, I. Peled, and O. A. Nielsen (2021). ""Is not the truth the truth?": Analyzing the Impact of User Validations for Bus In/Out Detection in Smartphone-based Surveys". In: *IEEE ITS Magazine (UNDER REVIEW)*.

Paper D: V. Servizi, D. R. Persson, F. C. Pereira, H. Villadsen, P. Bækgaard, J. Rich, and O. A. Nielsen (2021). "Large Scale Passenger Detection with Smartphone/Bus implicit interaction and Multisensory Unsupervised Cause-effect learning". In: *IEEE Transactions on Intelligent Transportation Systems (UNDER REVIEW)*.

Acknowledgements

To Francisco C. Pereira, Jeppe Rich, Niklas C. Petersen, Marie K. Anderson, Lucia Gaggiotti, Helle Marckmann, and Otto A. Nielsen for trusting the future.

To Mette N. R. Søndergaard, Leo A. R. Servizi, and Luca A. R. Servizi, for a time-warped present.

To my dearest family, friends, and colleagues, both standing and fallen, for a gifted past.

Valentino Servizi, December 2021.

Summary

Since their introduction, smartphones have constantly increased their market share. Smartphones allow users' identification, authentication, and billing. From a transport science perspective, smartphones can be used as a complex multi-sensor platform, enabling passive collection of human travel behavior. During this development, people and smartphones have become almost inseparable, especially during travel.

Smartphones can reveal new knowledge on transport behavior variations both between and within users. While traditional approaches are already measuring behavior variations between users, we need higher resolution to measure these variations within the same user. For example, one can alternate the use of bike and car according to weather conditions. For others, the alternation could derive from the day of week, the season, or the needs of some family members. On the one hand, cross-sectional interview-based surveys are unable to capture such details. Smartphones and their sensors may on the other hand offer unprecedented spatial and contextual resolution.

Handling such a higher resolution, however, provides a new complex set of challenges. Let us imagine a scenario in which smartphones and vehicles active on the transport network are continuously connected to the communication network for the purpose of providing an intelligent transportation service. The resulting data footprint of both sensors and algorithms would be huge. Moreover, the "intelligence" of such a system should follow each passenger on each instant of its journey. Regardless of the ongoing engineering challenges, mostly unsolved, learning people transport behavior in this scenario requires rich and efficient data representations, and knowledge of each trip's ground truth at the same scale as the data: this is in itself a gargantuan challenge, and this work moves initial steps to ease it.

The ability of measuring behavior variations within the same user could enable discoveries we cannot predict. Further, these measures may discover causal reasons for human transport behavior that existing measurement systems cannot provide. Although people travel only a fraction of the day, the purpose behind each trip is one of many activities defining their lives. The thesis contributes towards better measurements of the transport behavior.

For achieving significant measures of transport behavior variations within each user, while measuring variations between users, this Ph.D. thesis provides the following main contributions.

We pinpoint and examine the problems limiting prior research up-front. This step exposes drivers to select and rank machine-learning algorithms used for processing data generated by smartphones. It also shows the main physical limitations, and an overview of the methodological frameworks deployed for measuring transport behavior variations. The

output consists of a defined relationship among user interaction, methods, and data.

Next, we focus on two fundamental binary classification problems of Geographic Positioning System (GPS) trajectories. Both underpin many current and upcoming smartphone-based technologies deployed to measure transport behavior variations during a journey: one problem is stop-detection; the other is identifying users' presence inside or outside the transport network. Most of the problems relevant for detecting transport behavior variations belong to one or both of these two large categories.

In both cases the solutions share a framework of methodological-, technological-, and sensorial-information convergence. Solutions' quality affects directly the quality of transport behavior measures, such as inference of departure/arrival time, transport mode, trip purpose, and transit flows through the transport network.

For stop detection, we combine GPS time series with spatial context information retrieved from a Geographic Information System (GIS), which we represent as multi-dimension tensors. This line of work explores both simple and advanced data representations benchmarked through specialized artificial neural networks, random forest, and unsupervised machine learning baselines.

To classify whether one is inside or outside the transport network, we combine independent sensors measuring the same interactions between smartphone and infrastructure. We leverage signals allowing short-range implicit interactions between devices. To assess the potential, we verify how robust these signals and related machine learning classifiers are against the noise typical of realistic contexts.

We developed a proprietary smartphone-sensing platform collecting these independent and contemporary signals from devices installed on the infrastructure–buses in our use case– and Global Positioning System locations of both buses and smartphones. In a real experiment, we collected various levels of ground truth quality and smartphone-based sensors' data. Then we simulate human errors in the labelling process, as is known to happen in smartphone surveys when people validate travel diaries.

On large scale multi-modal deployments, widespread technologies sensing people presence within the transportation system–such as Implicit Walk-in/Walk-out (WIWO) and explicit Check-in/Check-out (CICO)–present limitations. For example, accuracy depends on the ground truth's reliability; scalability, on the sustainability of reliable ground truth. These limitations prevent Intelligent Transportation Systems from supporting analysis, optimization, and control of transport comfort, safety, and efficiency. Implicit smartphone-sensing aims also at closing this gap. We propose the Cause-Effect Multitask Wasserstein Autoencoder. This method acts as a powerful dimensionality reduction tool and obtains an auto-validated representation of a latent space describing users' smartphones within the transport system. Such a representation allows meaningful clustering, consistent with the

problem at hand, via DBSCAN (Density-based Spatial Clustering of Applications with Noise). Consequently, this method enables the output of ground truth at Big Data scale.

A general contribution we yield across the work presented above, stems from the ablation studies. Noisy signals affect the classification performance. However, the impact of this noise on the classification performance is not always intuitive. For example, let us consider a very noisy dataset. If this noise affects the signal, the classification accuracy computed after data cleansing would not consider a large fraction of the data lost in the cleansing step. If the noise affects the ground truth, false positives may be true positives and false negatives would be true negatives. Consequently, to support optimal decision-making, we propose two perspectives, which we introduce to complement metrics derived from the confusion matrix, such as Accuracy or F1-score. We measure the impact of noise for both GPS signal and ground truth. In the first case, we look at the correlation coefficient. In the second case, we simulate labelling errors. These measures of noise impact to the classification performance can be considered as key performance index, facilitating the comparison across different classifiers. Ground truth quality, as other signals, represents a random variable underpinning both the scalability and the performance of any classifier.

As a conclusion, the thesis provides the basis for methods enabling higher resolution measurements of human transport behavior variations at a Big Data scale, and the contributions mentioned above represent a promising step. The novel data structures and methodologies bring the potential of reduced bias in the measurements. At the same time, the impact of a reduced bias for methods' evaluation is direct and immediate.

Resumé (Danish summary)

Lige siden deres indtog på markedet har smartphones udvidet deres markedsandele. I løbet af denne udvikling er mennesker og smartphones blevet næsten uadskillelige, specielt under rejser. Smartphones gør det muligt at identificere og godkende brugeren samt opkræve betaling. Fra et transport-videnskabeligt perspektiv kan smartphones opfattes som en kompleks "multi-sensor" platform, der muliggør passiv indsamling af menneskers rejseadfærd.

Smartphones kan observere transportadfærdsmæssige variationer, både mellem brugere og hos den enkelte bruger. Traditionelle rejsevaneundersøgelser måler allerede adfærd, f.eks. den danske Transportvaneundersøgelse, der spørger til en enkelt dags rejse. Men der er brug for længere perioder og flere detaljer for at måle variationer hos den enkelte bruger. F.eks. kan en person veksle mellem brug af cykel og bil som følge af vejret. For andre kan denne vekslen skyldes ugedagen, årstiden eller hensynet til resten af familien. Tværsnitsdata er ude af stand til at indfange sådanne detaljer. I stedet skulle den samme person indrapportere sin adfærd over en længere, sammenhængende periode. Men det er typisk for krævende for respondenten. Med smartphones vil sådanne rapporteringer kunne gennemføres passivt (dvs. automatisk) og dermed tilvejebringe en hidtil uset nøjagtighed i forhold til rejseadfærds rumlige og tidsmæssige kontekst.

Sådanne data medfører imidlertid nye komplekse udfordringer. Lad os forestille os et scenarie, hvor smartphones og køretøjer, der er aktive i et transportnetværk, er konstant forbundne til et kommunikationsnetværk med det formål at tilbyde intelligent transport-service. Mængden af data fra de to systemer og algoritmer ville være enorme. Ydermere skal logikken i et sådant system følge hver passager hele tiden under hele rejsen, hvilket kræver en effektiv håndtering af store datamængder, og kontroldata indsamlet på anden vis (såkaldt "ground truth") til validering og konfigurering af algoritmerne.

Muligheden for at måle adfærdsvariationer hos den enkelte bruger kan lede til ny viden, og det vil kunne lede til indsigt i årsagssammenhænge i den menneskelige transportadfærd, som eksisterende målingssystemer ikke kan tilbyde. Selvom mennesker kun rejser i en lille del af deres dag, så er formålet med hver af deres rejser en af de mange aktiviteter, der definerer deres liv. Denne Ph.D.-afhandling giver følgende hovedbidrag i den henseende;

Afhandlingen udpeger og undersøger problemer, der ikke kan belyses med traditionelle typer rejsevaneundersøgelser. Til dette formål udvælges og rangordnes machine-learning algoritmer til at processe data fra smartphones. De vigtigste begrænsninger diskuteres, og der gives en oversigt over de metodiske rammer til at måle variationer i transportadfærd. Resultatet klarlægger sammenhængen mellem brugerinteraktion, metoder og data. Dernæst fokuseres på to fundamentale binære klassifikationsproblemer i Geographic Positioning Systemer (GPS); 1) detektering af stop på rejsen, og 2) identifikation af brugeres tilstedeværelse i eller udenfor transportnetværket. Størstedelen af de problemer, der er

relevante for at opdage variationer i transportadfærd, hører til i en eller begge af disse kategorier.

For detektering af stop kombineres GPS tidsserier med rumlige data om turenes omgivelser fra et Geografisk InformationsSystem (GIS). Dette repræsenteres i machine-learning algoritmen som multi-dimensionale tensorer. Der undersøges både simple og avancerede datastrukturer benchmarked gennem kunstige neurale netværk, random forest, og uovervågede machine-learning baselines. For at fastslå om en respondent er indenfor eller udenfor transportnetværket kombineres uafhængige sensorer, der måler de samme interaktioner mellem smartphone og infrastruktur. Vi udnytter signaler, der tillader short-range implicitte interaktioner mellem "devices". For at vurdere potentialet, testes hvor robuste disse signaler og benyttede machine-learning klassifikatorer er, set i relation til den unøjagtighed (støj), der er fra data.

Til testning af studiets teori, udvikledes en proprietær "smartphone-sensing" platform, der indsamlede uafhængige signaler fra enheder installeret på selvkørende busser på DTU Campus. Data omfattede Global Positioning Systems stedfæstelse af både busser og smartphones. Der indsamledes forskellige grader af "ground truth quality" og smartphone-baserede sensor data. Derefter simuleredes menneskelige fejl i "the labelling process", da det er kendt, at dette sker i smartphone-undersøgelser, når respondenter skal validere data.

I forhold til implementering i større skala, vil teknologier, der føler rejsendes tilstedeværelse indenfor transportsystemet – f.eks. Implicit Walk-In/Walk-Out (WIWO) og explicit Check-in/Check-out (CICO) – udgøre begrænsninger. For eksempel afhænger nøjagtigheden af "ground truth's" pålidelighed, og skalerbarheden afhænger af bæredygtigheden af pålidelig "ground truth". Disse begrænsninger forhindrer Intelligent Transportation Systems i at understøtte høj opløsning og storskala analyse, optimering og kontrol af transportkomfort, sikkerhed og effektivitet. "Implicit smartphone-sensing" sigter også imod at lukke dette hul, ved at benytte den såkaldte Cause-Effect Multitask Wasserstein Autoencoder machine-learning teknik. Denne metode virker som et kraftfuldt dimensions-reduktions-redskab og opnår en autovalideret repræsentation af "a latent space". En sådan repræsentation tillader en "clustering", via DBSCAN (Density-Based Spatial Clustering of Applications with Noise), der muliggør at etablere valide "ground truth" data i stor skala.

Studiet viste at støjende signaler i høj grad influerer på klassifikationskvaliteten på en måde, der ikke altid er intuitiv. Lad os for eksempel overveje et meget støjende datasæt. Hvis denne støj påvirker signalet, vil klassifikationsnøjagtigheden, der beregnes efter datarensning, ikke tage højde for en stor del af data, der er gået tabt i rensningstrinnet. Hvis støjen påvirker "ground-truth", kan falske positive være sande positive, og falske negative ville være sande negative. Derfor, for at forbedre kvaliteten af smartphone baserede dataindsamlinger foreslår afhandlingen to tilgange, der komplementerer målinger

afledt af "Confusion Matrix", som Accuracy eller F1-score. Påvirkningen af støj for både GPS-signal og "ground truth" måles. I det første tilfælde ses på korrelationskoefficienten. I det andet tilfælde simuleres "labeling" fejl. Disse målinger af støj-påvirkning på kvaliteten af klassifikationen kan betragtes som "Key Performance Index", der tillader sammenligning mellem forskellige klassifikationer. "Ground truth" er en tilfældig variabel, der underbygger både kapabiliteten og performance af enhver klassifikator.

Som konklusion giver afhandlingen grundlag for metoder, der muliggør højere opløsningsmålinger af variationer i menneskelig transportadfærd i stordataskala, og de ovenfor nævnte bidrag repræsenterer et lovende skridt fremad. De nye datastrukturer og -metoder giver potentiale for reduceret bias i målingerne. Samtidig er virkningen af en reduceret bias for metodernes vurdering direkte og umiddelbar.

Contents

1 Introduction

To measure user behavior variations in general and transport behavior in particular, engineers must climb a mountain, on top of which they will find at least philosophers, psychologists, and economists. The number of behavioral theories and accounts is very large in each discipline. Supporting these theories with objective and unbiased measurements, however, is not trivial. Rather than the behavioral aspects of transport, the scope of this work is on the measurements. Whereas most of the measures we use in our daily life underwent a complex and extensive standardization process, despite decades of intense work, some measures of transport behavior variations can be still considered in their infancy. However, both old and new measures have a lot in common: to be perceived they need a "sense" connected to a tool, and some harmonized methodology such that senses can be evaluated and then compared meaningfully. Next, our human senses need to be conscious of what these measures mean, because rational decisions are likely to be based on such measures, in this case to design and dimension optimal transport systems.

For example, looking at 60 years old Tokaido Shinkansen, the high speed train connecting Tokyo and Osaka in Japan, Takagi, 2005 argues that the introduction of new stops increasing of 6 minutes the total time between the two main station, from 2 hours and 30 minutes, contributed in increased ridership: Catching the train from intermediate stations, passengers perceived a door-to-door time reduction of the journey, which for many is more important than the travel time between the two main stations. Assuming the possibility of generalizing the positive impact of such a decision on a global scale, the value for passengers and society would be inestimable. To replicate, we need to improve our ability to understand what means value for passengers. The value of time is just one example (Ben-Akiva, 2017).

As Diane Ackerman writes in *A Natural History of the Senses*, *"To begin to understand the gorgeous fever that is consciousness, we must try to understand the senses–how they evolved, how they can be extended, what their limits are, to which ones we have attached taboos, and what they can teach us about the ravishing world we have the privilege to inhabit"* (Ackerman, 1991).

This thesis focuses on how to sense and then measure human travel behavior. We leverage smartphone and Internet of Things (IoT) as a sensing platform, and machine learning as the medium between artificial senses and human perception of the measures describing people travel behavior variations. *"The newest eyes are those we have invented"* (Ackerman, 1991).

Based on the prospect of global population in 2021 (United Nations DESA, 2019) and the number of smartphones subscriptions in the same year (Ericsson, 2021), an approximate

estimation shows that 81% of the population adopted a smartphone (in average across countries). With respect to the years 2015 and 2021 (Ericsson, 2021), while the number of estimated smartphone subscriptions for 2021 is confirmed–with a negligible overestimation of 1.5% in the forecast published in 2015–the total mobile traffic estimation required a significant correction, 27% above the forecast from 2015. In the same report, we find the perspective of a steady growth of smartphone subscriptions throughout the year 2027 with a yearly rate of 17%. This translates into a growth of smartphone subscriptions from 77% of all mobile subscriptions in 2021, to 86% in 2027. The pervasive market penetration of smartphone devices represents the unique opportunity to look at user travel behavior with an unprecedented resolution, because (i) smartphones embody a rich set of sensors on-board; (ii) smartphones rest on standards and protocols deployed to allow their interaction with the telecommunication network and the internet of things; (iii) smartphones' communication network and IoT surround users at any time, including while travelling; (iv) people carry their smartphones all the time, often even staying next to them when sleeping.

This chapter introduces the PhD thesis and unfolds the background on which the next chapters build upon. Each of the following chapters contributes to a specific aim within the transport context, casting a relationship between artificial senses, user behavior, and our "newest eyes": a higher-resolution detector invented to study people transport behavior.

1.1 Background

Transport services and infrastructure, both private and public, orbit around users' need of moving throughout the space to fulfil any required daily activity, for example, in relation to work, family, or other related tasks. Users' behavior and transport operations act and react across time, each under the strain of the other, within a dynamic demand and supply framework subject to multiple constraints and shocks challenging the optimal equilibrium of the system. Decisions of people, transport operators and policy makers can be modelled as the consequence of specific utility functions, having a very different sensitivity to the scale of the time variable. The time horizon affecting choices for these three players increases dramatically from the first to the last. Whereas people may choose a transportation to minimize travel time and transport cost for the next hour and the next trip, transport operators and authorities may choose investment and policies to minimize travel time and transport cost beyond several years of operations and for trips accounted at city or country scale. However, any player's decision affects on the system's equilibrium and contributes to the path between two consecutive equilibria.

Although users' behavior is extensively studied in Transport Economics and Psychology, and a large body of literature describes how different user profiles interact with the

transport system, to complement the theories with precise quantitative estimations, data scientists must pick-up the challenge left by statisticians, and climb the mountain where psychologists and economists stand. Random utility theory applied to transport (Train, 2003; Ben-Akiva and Lerman, 1985; McFadden, 1986) and transport models delivered as simulations of complex urban systems (Vuk et al., 2016), for example, rest on the behavioral measurements mostly taken with traditional travel surveys. Traditional travel surveys evolved over time to provide such measurements. For example, in Denmark, since 1975 the National Travel Survey (TU, Transportvaneundersøegelsen) collects data about travel behavior. The Centre for Transport Analytics at the Technical University of Denmark is now running the latest version of the survey, and started introducing a smartphone-based travel survey. To sustain statistical representativeness regarding the whole Danish population and keep it up to date, TU requires the collection of multiple new interviews every day of the year (Christiansen and Skougaard, 2013), totalling an average of 12000 interviews per year since 2010 (Christiansen, 2012). Each year this very representative cross-sectional sample, by definition, looks at only one day per respondent. For origin and destination of people trips, this method relies on what respondents say, and not on measurement devices. Few are the exceptions.

From paper-and-pencil to telephone-based and computer-based surveys to internet-based, the data collection process improved substantially. Nevertheless, through time, the scale of the data to be handled did not change significantly. The statistical methods developed to yield behavioral measurements from traditional surveys have been refined and perfected for such a scale and can handle exceptionally well cross-sectional surveys representative of the population subject to examination. From this standpoint, limitations arise on the maximum possible resolution of the picture describing variability between different user profiles. For example, while traditional travel surveys allow the definition of distinct user profiles and the difference between profiles is measurable, detecting the fluidity across profiles of the same user requires higher resolution and the ability of tracking multiple users, each for a long period. Traditional travel surveys can also do it, e.g., with a resolution of one interview per year, or a few interviews over a week. With smartphones, assuming continuous usage, the resolution would go down to the second and up to many weeks, months or maybe even years of data collected. This requirement suddenly changes the scale of the data to be handled, and if the smartphones represent the ideal platform to generate signals at this scale, translating these signals into measures is an entirely new challenge, full of potential.

1.2 Problem Definition

The higher-resolution detector allowing us to see high-resolution transport behavior includes artificial senses, machine learning algorithms, and evaluation methods consenting the optimal adaptation of this higher-resolution detector across different use cases. The

cornerstone signal comes from the Global Navigation Satellite System (GNSS) mostly known as the US-owed deployment named Global Positioning System (GPS), which we will refer to in the rest of this work. The GPS signal is extensively studied and describes the time series of a device location within the geographical space. In this field, GPS is the "king" of our artificial senses.

The literature exposed several shortcomings of the GPS signal. As opposed to open outskirt areas, signal noise and vehicles speed in urban areas represent two principal and synergistic components behind these limitations. In urban environments, excess of GPS noise or urban canyons affect devices' position with significant errors or "mirage-like" events, such as sleep walking (Yozevitch and Moshe, 2015), where a stationary GPS device appears to be moving in space, through time. Further limitations can apply to GPS sensors installed on smartphones, as varying devices quality, manufacture, hardware, and software may affect the variability of sensors' signal.

This work seeks to answer the following general research question.

> *To deliver a higher-resolution detector on human travel behavior variations–such as transport mode, trip purpose, and presence within the transport network–how can signals be combined and exploited to contrast the mirage-like limitations–also known as bias–of current behavior measurement methods?*

We leverage the very nature of GPS signal. GPS is a tuple consisting of longitude, latitude, and timestamp. Describing a device motion in space through time, GPS enables the augmentation of its description via other sensors or information's systems. The spatial-temporal position can bond with both the space and time signal that exists in each relevant domain, such as: geo-spatial information available in spatial proximity, existing in the space domain; sensors providing contemporary and independent signals as time series, existing only in temporal domain. This approach could fulfil more restrictive requirements.

Machine Learning (ML) represents another major component of the higher-resolution detector. ML is fairly very well studied for mining user behavior, e.g., from smartphones. In urban areas, different transportation modes travel at similar speeds. Consequently, transportation modes such as buses, cars, and bikes, present very similar patterns (Schuessler and Axhausen, 2009). Therefore, the correct identification of transport or route choices with ML is very challenging in urban and highly dense areas. Moreover, the identification process often depends on the ground truth underpinning supervised or semi-supervised ML training. Hence, since learned patterns could be attached to wrong labels, the identification process could be subject to a dangerous and hard to detect bias.

At larger scale, the aforementioned problems undermine the effective deployment of this higher-resolution detector and present the serious risk of wrong decisions leading to a

negative impact on, e.g., user experience through public transport network, environmental impact, time and resources. Whereas wrong decisions' impact on transport operations could be solved in a shorter time horizon, the impact on infrastructure would last much longer. Consequently, conspicuous economic, social, and environmental damages can never be expected at any small scale even in the best-case scenario. In worst-case scenario, negative impact of wrong decisions can be catastrophic. With this work, we want to reduce the incidence of wrong decisions due to wrong assumptions around sensors and ML methods involved in the study of users' transport behavior from smartphones, and the dwell about the interoperability of machine learning methods.

1.3 Contribution

To solve the problem as stated in the previous section and to enable a higher-resolution detector with the ability of measuring behavior variations both within the same user and between different users, we deliver four main contributions. All the contributions focus mostly on classification tasks.

Chapter 2 provides a self-contained and comprehensive literature review with multiple perspectives consolidated around tasks, technologies, methods, and data adopted to mining user transport behavior from smartphones. Existing literature and reviews provide deep and narrow perspectives. These seem unable to capture converging dynamics active across neighboring fields of research. In addition, standardization issues dwelling under the radar contribute to increasingly biased perceptions. We provide a logical connection between user transport behavior measures and the technology of the supporting smartphone-sensing-platform. The chapter presents the literature on how to combine methods and data streams to extract behavioral information. Ultimately, the review exposes converging methods and technologies applied to study independently various transport behavior aspects. Besides, we propose evaluation metrics supporting comparability across methods, at least to some qualitative extension: i.e., task complexity, method requirements, and dataset representativeness. We also show opportunities and threats present in the data validation process and in different interaction models: i.e., person-to-person, person-to-device, and device-to-device. Then we argue that deep learning and artificial neural networks can support strategic applications under-investigated in this field: i.e., model training standardization, data-fusion, and reduced dependency on labels.

Chapter 3 builds upon the opportunity of extracting higher resolution behavioral information by combining models, data structures, and signals. The work analyzes multiple artificial neural network models, deployed to classify whether a user is traveling or not on the transport network. The chapter focuses on point-based classification of GPS fused on Geographic Information System data, resulting in multi-dimension tensor representations. We perform an ablation study and compare other popular supervised and unsupervised

classifiers. This study finds that the proposed model performs substantially better, in particular when GPS trajectories are affected by noise.

Chapter 4 builds upon the opportunity of supporting user behavior measurements systems at scale, evolving from person-to-device towards device-to-device validation of the measures. The work includes design and deployment of a sensing platform for smartphone's on-board sensors, including Bluetooth Low Energy (BLE) devices in its proximity. This platform collected and stored data describing user transport behavior variations of a number of traveling users. The work focuses on the binary task: the classification of whether users are inside or outside the transport network. The use case includes autonomous buses operating on a simple but realistic transport network. The special setup allowed video recording of each user's trips through this network, which represent a high quality ground truth. Ground truth was also collected with a person-to-device interaction, directly from users. The dataset comprehends the native iOS and Android classification of transport mode, based on the time-series of the smartphones' inertial navigation system (INS). We compare models performance based on realistic INS-, GPS- and BLE data. The models were trained and/or evaluated using various levels of ground truth quality: i.e. flipping-labels, with various noise levels on recorded trip time-series, obtained through the simulation of human labeling errors. In this full-stack project, we exposed the impact of noisy labels in the training and evaluation of supervised models. With respect to the literature reviewed in this field, results show that good quality ground truth should not be just assumed, but should be tested to avoid dangerous biased perception of the models' performance.

Finally, Chapter 5 builds on the same task and data structure described in Chapter 4, and extends some of the artificial neural networks models presented in Chapter 3. To guarantee extreme scalability potential, this work presents a lightweight and novel powerful unsupervised classifier, based on the intuition that redundant and independent sensors' signals may substitute labels, and thus enable device-to-device interaction for user's behavior auto-validated measures. We combine and extend several artificial neural network frameworks, we perform an ablation study with multiple processing architectures, and we compare the performance against the best available supervised classifiers. In the comparison, we also introduce the analysis on models' sensitivity to labels' noise. Results show that under the assumption of optimal ground truth, our solution is comparable to the best-supervised classifiers. In presence of noise in labels, our solution outperforms all the baselines. This property is crucial to support service disruptions, such as route change due to roadwork or traffic congestion, which require re-training the models, in this case with no labels.

The datasets underpinning Chapters 3, 4 and 5 represent users moving exclusively through urban and high-density areas, located in Copenhagen. The same conditions were recreated while collecting the dataset underpinning Chapter 4. Consequently, we always deal with

at least two constant challenges: low speeds, and relatively tall buildings in proximity. This is seldom the case for the literature reviewed in Chapter 2, where datasets seem to contain trajectories collected on both urban and outskirt areas.

1.4 Outline of The Thesis

Following this introduction, Chapter 2 presents a review of sensors, datasets, features, and methods designed to mine user transport behavior from smartphones. The focus is on Smartphone-based travel surveys. Chapter 3 focuses specifically on one perspective: modelling artificial neural network for stop detection of GPS signal, fused on the geo-spatial domain with data from Geographic Information Systems. Chapter 4 focuses specifically on analyzing the impact of noisy labels on training ML models, and assesses GPS and BLE signals. Chapter 5 builds upon the previous Chapters, and focuses on modelling artificial neural networks for detecting users' presence within the transport network, based on capturing the behavior variations sensed independently from GPS and BLE signals. The architecture of this neural network leverages the cause/effect relationship between the two independent signals, and avoids the correlation. Final remarks and future perspectives conclude this work in Chapter 6.

References

Ackerman, D. (1991). *A natural history of the senses*. Vintage.

Ben-Akiva, M. (2017). "Choice Modelling: The State-of-the-art and The State-of-practice". In: *Proceedings from the Inaugural International Choice Modelling Conference*. Ed. by S. Hess and A. Daly. Emerald. DOI: 10.1108/9781849507738.

Ben-Akiva, M. and S. R. Lerman (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA, USA: The MIT Press.

Christiansen, H. (2012). "Documentation of the Danish national travel survey". In: *Lyngby, Department of Transport, Technical University of Denmark*.

Christiansen, H. and B. Z. Skougaard (2013). "The Danish National Travel Survey-declaration of variables TU 2006-12, version 2: Documentation note". In.

Ericsson (2021). *Ericsson Mobility Report*. Online; accessed November-2021. URL: https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2021.

McFadden, D. (1986). "The choice theory approach to market research". In: *Marketing science* 5.4, pp. 275–297.

Schuessler, N. and K. W. Axhausen (2009). "Processing Raw Data from Global Positioning Systems without Additional Information". In: *Transportation Research Record* 2105.1, pp. 28–36. DOI: 10.3141/2105-04. URL: https://doi.org/10.3141/2105-04.

Takagi, R. (2005). "High speed railways: the last 10 years". In: *Japan Railway and Transport Review* 40, pp. 4–7.

Train, K. E. (2003). *Discrete choice methods with simulation*. DOI: 10.1017/CBO9780511753930.

United Nations DESA (2019). *World Prospect*. Online; accessed November-2021. URL: https://population.un.org/wpp/Graphs/DemographicProfiles/Line/900.

Vuk, G., J. L. Bowman, A. Daly, and S. Hess (2016). "Impact of family in-home quality time on person travel demand". In: *Transportation* 43.4, pp. 705–724. DOI: 10.1007/s11116-015-9613-2. URL: https://doi.org/10.1007/s11116-015-9613-2.

Yozevitch, R. and B. B. Moshe (2015). "A robust shadow matching algorithm for GNSS positioning". In: *Navigation: Journal of The Institute of Navigation* 62.2, pp. 95–109.

2 Paper A: Transport behavior-mining from smartphones: a review

The following pages contain the article:

V. Servizi, C. F. Pereira, K. M. Anderson, and A. O. Nielsen (2021). "Transport behavior-mining from smartphones: a review." In: *European Transport Research Review*. DOI: 10.1186/s12544-021-00516-z. URL: https://doi.org/10.1186/s12544-021-00516-z.

Please cite accordingly.

The work was presented at the seminar "10 years of preparation for smartphone-based travel surveys: what are the opportunities, barriers, recommendations, and good practices to drive the switch-over?", *TECHNION-Israel Institute of Technology*, Haifa, Israel in December 2020.

Part of this work was presented within the "GNSS-based Classification Including Emerging Two-wheels Electric and Shared Transport Modes, with Semi-supervised Artificial Neural Networks", at the *1st Israeli Smart Transport Research Center Annual Conference*, Tel Aviv, Israel, in June 2021.

Part of this work was also presented within the "Travel Mode Detection using Semi-supervised Artificial Neural Networks and Transfer Learning", at the *5th Cycling Research Board Annual Meeting*, Copenhagen, Denmark in October 2021.

Transport behavior-mining from smartphones: a review

Valentino Servizi*, Francisco C. Pereira, Marie K. Anderson, Otto A. Nielsen

Department of Management Engineering
Technical University of Denmark (DTU)
Kgs. Lyngby Denmark

Abstract

Although people and smartphones have become almost inseparable, especially during travel, smartphones still represent a small fraction of a complex multi-sensor platform enabling the passive collection of users' travel behavior. Smartphone-based travel survey data yields the richest perspective on the study of inter- and intra-user behavioral variations. Yet after over a decade of research and field experimentation on such surveys, and despite a consensus in transportation research as to their potential, smartphone-based travel surveys are seldom used on a large scale. This literature review pinpoints and examines the problems limiting prior research, and exposes drivers to select and rank machine-learning algorithms used for data processing in smartphone-based surveys. Our findings show the main physical limitations from a device perspective; the methodological framework deployed for the automatic generation of travel-diaries, from the application perspective; and the relationship among user interaction, methods, and data, from the ground truth perspective.

Keywords: Smartphone-based travel surveys, machine learning, user behavior, transport, map-matching, mode detection, activity inference, data fusion

*Corresponding author. Email: valse@dtu.dk

1. Introduction

To support the planning, design, and policy-making processes for improving transport systems [1], travel surveys capture essential aspects of user behaviors on which behavioral modeling relies [2]. For designing the representativeness of a user sample under study, the statistical approach in traditional travel surveys is prominent. The process involves person-to-person (P2P) interactions for data collection, a process overlapping with ground truth collection: Trained travel surveyors directly validate data with users and manually reconstruct users' travel-diaries for behavioral study.

In contrast, machine-learning plays a primary role in smartphone-based travel surveys (SBTS). The data collection process involves device-to-device interaction, with machine-learning algorithms automatically reconstructing users' travel-diaries directly from data that might contain various sources of errors [3]. By submitting each travel-diary to the user for validation (i.e., to find out whether the user needs to change the travel-diary or not), the process can collect ground truth through a person-to-device (P2D) interaction between the user and an input/output interface, either via a website or smartphone [4].

Since the introduction of the first generation of smartphones equipped with assisted global positioning systems (AGPS) in the early 2000s, researchers have described smartphone-based travel surveys as a promising platform to measure user transport behavior. They can track the same user with an extended time horizon [5], collect data passively [6], detect previously unreported short trips, and avoid stereotypes of daily activity [7] (e.g., "I don't remember what I did, but here's what I usually do"). Given that SBTS would likely facilitate the discovery of inter- and intra-user behavior variations, the question is why SBTS have not yet replaced traditional travel surveys [8].

For researchers and public authorities, standardized performance indexes based on standard datasets support optimal investment decision-making. This approach also applies to classification or regression methods underpinning the identification of user transport behavior variations. Nevertheless, standardization in this field is lacking. Instead, decision-making often relies on assumptions, such as (i) consistent performance indexes evaluation across studies; (ii) comparable performance indexes across studies, even when based on different datasets; (iii) adequate representativeness of the few public datasets available; (iv) exact ground truth. By definition, each necessary

assumption represents a knowledge gap.

We ask and answer the following questions: What are the main machine-learning methods that are used in the field? What is the relationship between ground truth and machine-learning methods? What are the primary datasets studied? What characteristics do these datasets have, and what features can we extract from them, and how? What are the challenges for machine-learning in the field of SBTS? What are the main implications for transport science?

To tackle these questions, we proceed by snowballing first forward and then backward [9]. We cover deterministic and machine-learning methods based on different datasets collected from across the world. We examine how models and algorithms exploit various data sources such as AGPS, Inertial Navigation Systems (INS), Geographic Information Systems (GIS), and Internet-of-Things.

The paper analyzes technologies enabling SBTS data validation, such as data preparation and feature extraction, and focuses on machine-learning methods for mining user's behavior from smartphone data. These methods target why people travel, where along the transport network they travel, and which mode of transport they use. These technologies make an impact by reducing resources associated with running traditional travel surveys, while enhancing users' transport behavior data-resolution. Following this approach, we are able to review purpose imputation, map-matching, and mode detection methods.

Existing literature and reviews offer a clear picture of how algorithms and background technologies evolve to provide improved measures of users' travel behavior variations. For example, we list several specialized methods with impressive performance scores. We also find unilateral perspectives offering standardization pathways for both methods application and performance evaluation. In practice, limitations such as data representativeness, ground truth quality, and performance evaluation procedures may often result in a biased perception of each method's potential.

Decisions based on wrong assumptions and biased perceptions represent a threat to the progress of this field. To bridge the gap, we provide the following contributions. We deliver a self-contained overview connecting the user transport behavior measures with the supporting smartphone-sensing-platform. We detail how available methods can be combined to extract behavioral information from various data streams. We show the convergence between research areas studying complementary aspects of transport behav-

ior. We organize each reviewed work by task complexity, method requirements, and dataset representativeness. So we facilitate methods' assessment and comparison across specific use cases, mitigating the limitations of dry and incomparable performance scores. The paper reveals opportunities offered by device-to-device interactions for data validation instead of other interactions, and exposes gaps in deep learning strategic applications.

The first section below presents the dimensions describing transport behavior and the tools embodied in a smartphone device for data collection. The following section describes the methods used to identify transport behavior from data and an overview of the implications for transport science. The subsequent discussion presents a joint look on the results of the surveyed literature, which the conclusion summarizes from a big-data perspective. At the end, we include the Tables organizing the main features of the literature reviewed.

2. Measures and Tools

To support the reader through the following analysis and discussion, we start by providing context and presenting concepts on which the paper rests, i.e., definitions, employment, and technological framework of SBTS.

2.1. Measures of Transport Behavior

The following terms are used to describe a user's journey (throughout a single day, for example; see Fig. 1) and represent the different variables, or measures, that SBTS is used to collect for studies on transport behavior.

Tour. Aggregation of trips, such that users' travels start and end at the same place, e.g. at home [10].

Trip. Travel entity identified with a set of attributes such as: start-location, start-time, purpose, transport mode, arrival time, arrival location [10].

Leg. Also identified as a "trip segment," this is the unimodal segment between two stops. Each trip segment has a start-time and -location, end-time and -location, and stop-purpose at the end of the leg (see Fig. 1, B) [10, 11].

Purpose. This represents what triggers the trip from origin to destination (see Fig. 1, A, C, D), and identifies the "activity" performed at the end of a trip.

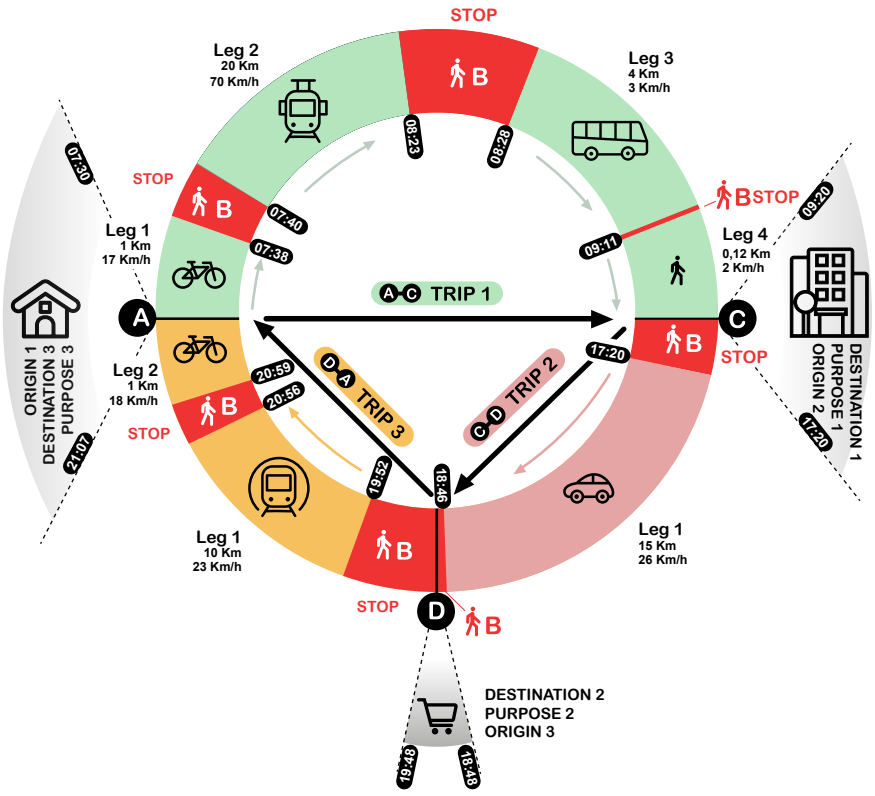

Figure 1: Tour Components and Travel Diary.

Stop. This can be reduced to two categories: stops at the end of legs (see Fig. 1, B), and stops at the end of trips (see Fig. 1, A, C, D).

Transport mode. This refers to a trip leg [12] and identifies, e.g., walking, cycling, car, train, bus, light rail (see Fig. 1).

Mode-chain-type. The literature provides no strict consensus on the definition of this term, and we define it as the list of transport modes one uses to get from the origin to the destination of a trip (see Fig. 1).

Travel-diary. This can focus on "one-day" (see Fig. 1) or on "multiple-days" and it describes the user trips through: (i) legs, where each leg has a unique transport mode; (ii) purpose; (iii) stops; and (iv) mode-chain-type. Generally, it is linked to a user, and his or her link-able personal information, such as: (i) age; (ii) occupation; (iii) education level; (iv) home address; and (v) work address. [10] presents a detailed list of further personal attributes.

Ground truth . This describes the true measurements of the target variables, for example the purpose of a trip, its transport-mode-chain, and the route between origin and destination. In general, the literature refers to (i) travel-diary; (ii) prompted recall survey; (iii) user input in mobile phones [13]; (iv) experiments (e.g. mode known) [14]; (v) trips reported in-situ by the user participating in an experiment [15]; and (vi) "traffic counts" extracted from video recordings [16]. However, because ground truth is lacking in several studies [17], authors have introduced alternative methods to close this gap, the results of which serve as a benchmark [18]. In case of synthetic data, studies on map-matching refer to the random selection among a set of alternative shortest paths [19]; in case of real data, other studies refer to GPS receivers collecting two independent measures, where ground truth is the measure with a higher sampling rate [20]. When algorithms target public transportation, ground truth can be extracted as the combination of bus stops and intersections within the transport network [21]. In the best-case scenario, the information is reported by users. As ground truth always seems prone to errors, [22] have introduced the concept of "*acceptable truth*," which, while not truly absolute, may be considered sufficiently accurate relative to the application.

2.2. Pioneering Smartphone-Based Travel Surveys

Within the last 20 years, traditional travel survey methods have been subject to the pressure of disruptive technological evolution. The large penetration of smartphone devices equipped with low-cost sensors, the introduction of Web 2.0, and the emergence of other directly related phenomena, such as

Big Data [23], could represent a tipping point for this research method [6]. There are several reasons to complement and/or substitute traditional travel surveys with smartphone-based technology, given the former's shortcomings, as follows:

1. Statistic representativeness, improvable or decreasing in some population's strata [24];

2. Trend of unreported short trips which the user tends to forget or does not want to mention [7];

3. Undetected behavior variations of the same user, due to the design of traditional travel surveys, which collects a cross-section sample of the population by focusing on one single day for each respondent [5];

4. Data collection cost per surveyed user [25].

The first large-scale SBTS deployments were the Future Mobility Sensing (FMS) in 2012, and the Sydney Travel and Health Survey in 2013. Most of the SBTS we know offer either web or app validation (seldom both), use machine learning, and are fully automated, as for example: (i) FMS/Mobile Market Monitor [26]; (ii) TRAVELVU/Trivector [27]; (iii) RMOVE/RSG [28]; (iv) Itinerum [29, 30]; (v) MEILI [22]; (vi) Sydney Travel and Health Survey [31]; (vii) Dutch Mobile Mobility Panel [7]; and (viii) MTL Traject [32].

These SBTS no longer collect ground truth via person-to-person interaction. Instead, their interfaces provide users with options to validate travel-diaries accurately generated, and to correct errors of the inaccurate ones, collecting ground truth via person-to-device explicit interaction. Nonetheless, users seem unable to report inaccurate diaries that are too difficult for them to correct on their own [33]. Consequently, the risk of encountering incorrect data within ground truth seems unavoidable for survey data. Regardless of whether available ground truth is acceptable or inaccurate, it is important to assess each application on an individual basis in the context of field research.

Success depends also on users' willingness to keep such an application installed on their smartphones. The main drivers determining the decision of a user to keep applications on his or her device are: (i) The information conveyed through the App; (ii) ease of use; (iii) perceived usefulness; (iv) perceived risks; and (v) general satisfaction of the user experience [34].

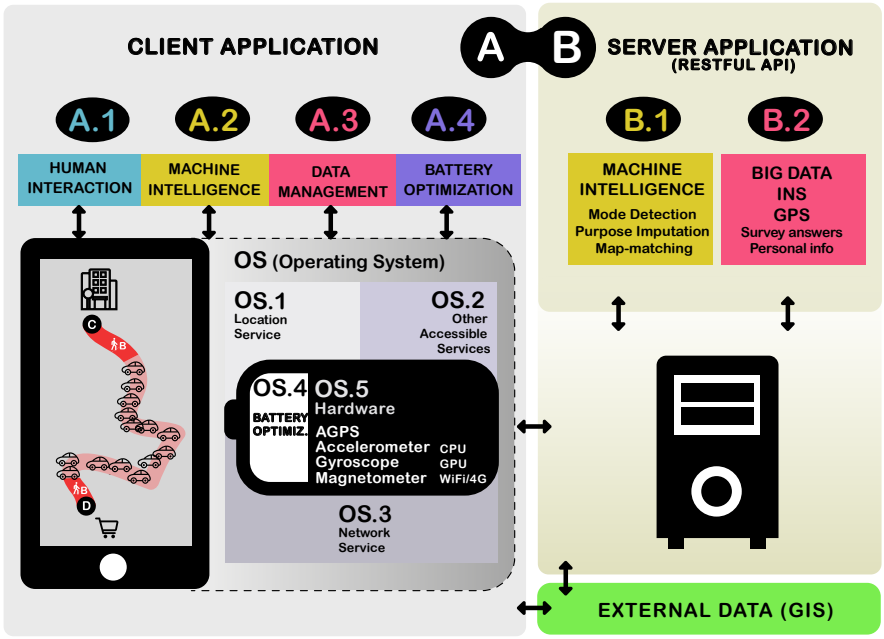

Figure 2: Smartphone-based Travel Survey Platform Architecture.

In (v) we mention a broad and very relevant field of research in which there is consensus about the negative impact of smartphone battery consumption on the user experience, which affects applications' penetration and drop-out rates. Because of the impact on quality of data collection, we observe the same consensus on battery concerns in the field of SBTS [24]. Also, the need of high resolution data in SBTS clashes with the need for battery efficiency enforced by smartphone platform providers [35].

Due to the highly-accurate trajectories generated by smartphones (e.g., through AGPS) and used by SBTS researchers, users are concerned by the potential for privacy violation. These trajectories often expose very personal information of each surveyed user, thereby presenting new challenges [36] in terms of reconciling a need for high-resolution data and a need to ensure privacy for researchers and users, respectively [37, 38].

2.3. Smartphone Capabilities

In Fig. 2 we present the abstraction of an SBTS platform. The main platform's components are client and server. The client (see Fig. 2, A) enables human interaction, e.g., for user travel diary validation (see Fig. 2, A.1), and orchestrates sensors, user-generated data (e.g., location), and computer intelligence models. Processing data locally, the client prevents loss of information, and maximizes privacy (see Fig. 2, A.3). A battery efficiency layer tunes and optimizes, e.g. data sampling or network input/output operations among client, server, and external data sources (e.g., GIS).

The sensory system of the platform is the smartphone, represented by:

- Principal hardware components (see Fig. 2, OS.5);
- Services exposed by the Operation System (OS, see Fig. 2, OS.1-OS.3); and
- Operations beyond users and developers influence, such as those focusing on device battery life extension (see Fig. 2, OS.4).

The following list of components is ranked by highest battery consumption to lowest [39], [40]:

1. Graphical Processing Unit (GPU) and screen, triggered when users interact actively with SBTS (e.g., validating travel-diaries).
2. Central Processing Unit (CPU), engaged also by computer intelligence models for online mode classification, for example, and for detecting conditions to switch off unnecessary sensors. While computation offloading to a server is possible, it implies transmitting data at its own energy cost.
3. AGPS. While GPS depends exclusively on satellites, in smartphones AGPS uses internet to look up the position of satellites and mitigate the cold-start problem. AGPS also uses cell-tower data. This feature is convenient when GPS signal is weak or disturbed, but it introduces challenges for position accuracy. To provide the location of a smartphone while reducing AGPS up-time, several effective strategies are available [41]. Finding the best trade-off between location accuracy, data resolution, and energy consumption is not trivial. Interestingly, we observe a convergence between approaches developed for the OS

to improve the energetic efficiency of smartphones, and for datamining to fill data gaps resulting from missing or highly uncertain GPS observations. Both provide location coordinates, reducing GPS sensor need, and leveraging data from INS, GIS, and telecom networks. Nevertheless, some of the current smartphone operation systems do not allow direct access to telecom network data from independent applications [42].

4. Network. An efficient tuning should consider network selection (Cellular or WiFi), data transfer frequency, battery status, and size of the data-transfer.

5. Accelerometer, gyroscope, and magnetometer raw data is accessible on the main OS platforms. GPS up-time is often optimized by leveraging these sensors to detect whether a user starts or ends a trip [41]. In general, accelerometer and gyroscope readings from smartphones should be collected with a resolution compatible with the motion frequency of human bodies in daily routines, which is above 20 Hz [43]. The consumption of such high-frequency data streams within the device is not critical for the battery. However, in case of transfer for storage and data consumption offline, handling the number of sensors and the high frequency quickly become critical for the smartphone's battery and for the user's data plan.

Sensors up-time and data transfer to the back-end, as well as the Ground Truth collection on screen are very critical for smartphones battery life [44]. For example, given a fixed data sampling rate, AGPS battery consumption is relatively more sensitive to the up-time, while high frequency sensors consumption is relatively more sensitive to data transfer. If not properly handled within the SBTS, battery drain could occur twice as fast, limiting the battery life to few hours instead of the whole day. Consequently, the impact of service interruptions would result in increasing limitations on the data. Covering the entire day for certain users would no longer be possible, and such a negative user experience would even increase risk of drop-out [34].

2.4. Physical Limitations for Data Validation

In addition to the aforementioned battery consumption issues, further critical implications of moving to this new technology are presented below.

2.4.1. Person-to-Device Validation

Design simplicity and intuitiveness should reduce any potential to distract the user while interacting with the survey application, as distractions could impact the quality of ground truth collected [24]. Furthermore, when the purpose of the interaction is directed to amend inaccurate travel-diaries, the impact that the design has on the quality of the ground truth collected from the respondents is even greater. A poor interaction between users and an SBTS interface could trigger a critical loop in which users validate wrong predictions instead of correcting them [45, 46].

2.4.2. Device-to-Device Validation

Arising from the convergence of Bluetooth and WiFi protocol in the Internet of Things context, and unlike the classic Bluetooth protocol, Bluetooth low-energy beacons communication is one-to-many (as traditional television or radio), involves few bits of data to be broadcast frequently, and needs no pairing operations. These properties are suitable for proximity detection and interaction with smartphones, and for activity sensing [47, 48]. A pioneering device-to-device ground truth collection on bus trips [49] already experimented Bluetooth low-energy interaction with SBTS, as an independent and redundant measurement of users' bus trips. This system has the potential to release users' resources that could the be exploited, for example, for filling in context-specific active surveys, and not for validating a travel diary. However, the authors highlight the challenge of finding a signal strength that allows for smartphones to detect beacons in conditions where signals may be attenuated or interfered with. A user's body or location, for example, may attenuate a signal, while interference with other beacons in range could result from passing by a bus stop or grouping with other buses.

3. Measuring Transport Behavior

The primary objective of SBTS consists of accurate ground truth collection from surveyed users. The correct reconstruction of travel-diaries, which encompasses both the transport mode and the purpose of any trip, allows for this goal to be achieved. Research on transport behavior also studies trajectories generated by the same sensors mentioned earlier. Therefore, it applies the same methods described in the following sections. In contrast with SBTS, however, research on transport behavior has the main objective of analyzing behavior, and not of collecting trip ground truth. This subtle difference may

support the large community of researchers claiming that mode detection methods should be agnostic to personal and location context (see Tables 1, 2 and 3). For example, the same method could generally serve different mode choice studies across the globe. In SBTS, this constraint does not seem to hold since travel-diaries also require predicting each trip's purpose, relying on both sensors and geospatial information (see Table 6). Successful hybrid approaches in this field further expose the shortcomings of such a purist approach. Data preparation is propaedeutic for learning the mode, purpose, and route of any trip. Simultaneously, cross-field convergence proves to be effective; for example, mode detection improves map-matching [17] and purpose imputation tasks [50, 12]. Inversely, map-matching GPS trajectories upfront improves the mode detection task [51]. When outputting a travel diary that allows ground truth collection on users' journeys, we do not find advantages from self-imposing restrictions on what data we should use or what method we should combine. Therefore, we find it beneficial to review purpose imputation and map-matching methods in this context. Tables 4, 5 and 6 present purpose imputation; Tables 7, 8 and 9 map-matching methods.

3.1. Smartphone Data Mining

Due to the disparity of progress drivers, we see a trend of increasing fragmentation, inconsistencies, availability, and volume of travel data. In response to this challenge, two main branches seem to arise as flip sides of the same coin [52, 53, 54, 14]. The first focuses towards data fusion, intended to compose and then mine high dimensional datasets collected from multiple sources, including GIS, INS, and GPS. The second targets the development of, for example, very sophisticated computer intelligence models, feature extraction methodologies, and optimal hyper parameters selection. These are constantly improving and therefore complementing traditional statistical methodologies, often substituting them for specific purposes [4].

Literature has shown that smartphone data is affected by several errors. For example, map-matching observations based on positions generated by a Nokia N95 would be much less reliable than those based on a dedicated GPS logger [55]. With current smartphones, however, the situation has improved substantially. For mode detection, neural network classifiers have shown higher performance on data collected from smartphones than from GPS devices [56]. Nevertheless, we should be aware that raw sensor measurements may vary between smartphones, as well as within the same model

of smartphone [57]. Any measurement is affected by noise that is not necessarily random, since it may be correlated with: weather conditions; building density, materials, and height; crowdedness; physical placement of the smartphone (e.g. in the pocket is different than on a table); smartphone model; and software "bugs." Therefore, achieving consistency of machine-learning methods across different smartphones requires a rigorous process of data preparation, cleansing, and trajectory segmentation up front. We describe these processes in the next sections.

For each classifier, such as for mode detection and purpose imputation, the underlying features can be (i) location-agnostic versus location-specific; and (ii) user-agnostic versus user-specific. For example, methods relying on user- or location-agnostic features can be trained on any geographic area, and then either deployed on a different area to classify the activities of another population or reused to solve similar problems. The former depends on the generalization power of the model, while the latter is identified as transfer learning. Transfer learning is the discipline dedicated to using the knowledge gained by solving a problem in one domain (e.g. stop detection) to solve a different problem in another domain (e.g. mode and purpose classification). From our standpoint, these approaches could contribute in mitigating the cold-start problem [58], for example in the process of switching from a traditional to a smartphone-based travel survey.

The literature reviewed often works with location and user-agnostic features. In contrast, user- [26, 59] and location-specific [11] data seem to enable more accurate classifications. Although results presented in the relevant literature are hardly comparable across studies, within each relevant study we find evidence about the positive contribution of user- and location-specific data on the performance of the classifiers [7]. The cost is the volume of information to be handled, poor transferability and poor generalization power. From this angle, we challenge the conclusions of [60]: Transferability and generalization power may also be related to the supporting dataset, and not only to the machine-learning method.

3.2. Data Cleansing

While performing data cleansing, data analysts should check whether basic features such as speed and acceleration are consistent with the context. The data cleansing purpose is to find and remove outliers, fill observation gaps, and possibly smooth the trajectories [61]. This crucial step should begin performing a sanity check on the observations' timestamps. Common

issues are multiple observations with the same timestamp, or discrepancies due to implicit time localization that keeps no trace, e.g., of periodical solar and legal time shifts. The first case can be mitigated using fine grained timestamps during data collection, such as milliseconds or microseconds; the second, using standard date representations such as the ISO 8601. Further, sensors trajectories are often stored inconsistently on database, e.g., due to smartphones temporary lack of internet connection. Therefore, to find "correct outliers", any basic feature—such as speed, space, and time variation between consecutive pairs of observations—should be computed after sorting these trajectories by timestamp. Once the basic features are available, to handle outliers there are different degrees of sophistication between rule-based, statistical, and model-based filters, such as threshold-, median-, and Kalman-filter. The measurements' sampling rate is a critical factor determining the filter choice. In general, the trade-off is between scalability and accuracy, with rule-based filters on the one hand, and more sophisticated tools like the Kalman-filters on the other. If the number of outliers is very high, such that removing these outliers we create unacceptable gaps in the trajectories, data analysts can resort to one of the several data imputation techniques available [62], such as an exponential weighted moving average.

To reduce the risk of noisy labels that could bias supervised classifiers already in the training phase, data cleansing should focus on labels too. Often labels come as a separate trajectory, which should have a common timeline with the sensors' observations. We are aware that during the validation users may overlook errors present on travel diaries. We cannot exclude human-computer interaction problems facilitating human errors during the travel diary validation step. Human errors may also occur while extracting data from the database. Rather than outliers, in these cases we should be concerned of flipping-labels [63]. Given a set of labels that a travel survey collects, outlying-labels indicate one or more trajectories labeled with a class not included in this set; flipping-labels indicate one or more trajectories belonging to one class and labeled with another class, both being present in the set. However, while the impact of both outlying- and flipping-labels on supervised classifiers is extensively studied for independent and identical distributed data [64, 65, 66, 67, 68]—for example on the popular handwritten digits dataset from the Modified National Institute of Standards and Technology database—we found no literature focusing on time series, as for example GPS.

3.3. Trajectory Stop Detection

The analysis of human trajectories can be reduced to two fundamental classes: **motion**, and **stop**. Tables 1 and 4 present how each class branches out. Tables 3 and 6 specify both features and methods enabling accurate classifications. Tables 2 and 5 present the dataset that enabled each study we reviewed. To perform any specialized inference on trip legs we need to identify homogeneous segments and relevant discontinuities from heterogeneous and complex mode-chain-types.

A GPS segment is considered a stop candidate if it lays within a topologically closed polygon for a certain time [69, 70, 71]. The presence of GPS points nearby may be indicative of a stop—the absence of motion [72]. Rules to acquire a local density of points, for example, include a moving window linking 30 preceding and 30 succeeding points within a 15 m range [73]. Although compatible with the error amplitude of GPS devices declared in a survey by [74], this range seems too small compared to smartphone AGPS expected error [75]. Smartphones location output does not rely exclusively on GPS, but also on less accurate methods that fill GPS gaps. [26], for example, extend the range to 45 m.

Based on the assumption that noise detected in transition points is temporary while the changes in speed are permanent, affinity propagation clustering methods can be effective in stop detection [70]. By building a network that links stationary events, identified as nodes within a critical space-time range, and clustering this network using two-level Infomap [76], a swift algorithm, available as python package [77], outputs a label for each stop event detected in a raw GPS trajectory.

Literature shows many developments in this direction, employing clustering techniques [78, 79, 80, 81], which can learn in an unsupervised fashion and find stops within GPS trajectories. In multiple-step approaches, personal- [80], and geographical-context [79] can augment trajectories' information and improve the classification of stop candidates. Density-based spatial clustering of applications with noise (DBSCAN) is at the base of most frameworks; some of these frameworks can even find stop candidates directly on raster image representations [82]. Many other effective probabilistic unsupervised methods are available, as for example kernel-based [83, 84]; generative [85, 86]; and discriminative [87], such as kernel-density algorithms, Hidden Markov Models, and conditional random fields.

Assuming that travelers walk to change mode, a rule-based algorithm can identify transition points by applying thresholds on speed, acceleration,

range and time, as well as by checking GPS on-off status [51]. In fact, the most common rule-based stop detection techniques rely on range, time, speed or acceleration thresholds [14].

These rule-based algorithms can be further improved by statistical tests. For example, a Kolmogorov-Smirnov test on a random sample can be used to check for outliers [88], as the normal distribution is sometimes accepted as a suitable approximation for GPS. Assuming normal distribution of GPS error, though, GPS follows a bi-variate Raleigh distribution [55].

Rule-based algorithms are both effective and appropriate, and are independent of the subsequent classification task, as for example mode detection, or purpose imputation. However, thresholds inflexibility (for example, in handling GPS signal loss and signal noise) leads to poor performance in detecting short stops (such as alighting from a bus) and long permanence in the same position (such as sitting on the bus during and intermediate stop) [14].

3.4. Trajectory Segmentation

Another approach specialized in "mode detection" is a GPS trajectory preparation through segmentation, which goes through four steps [89]. The first step splits the trajectory in fixed segments having the same size of the median number of points on all the available trips. The second step concatenates together consecutive segments with the same label. Let us note that the first two steps depend strictly on the availability of the ground truth, while the segment size depends on the data collection context. The third step discards segments with less than 10 GPS points. The fourth step smooths the trajectory through a Savitzky-Golay filter.

Segmentation methods can be distance-, time-, bearing- and window-based. While the last three are statistically equivalent, the first leads to varying sample sizes within each segment due to the different speeds in complex mode-chain-types. Discontinuities in the mode-chain-type, detected on these segments, represent stops [90].

The impact of stop-detection or trip segmentation on the quality of the travel diary generation process, and therefore on the quality of the ground truth collected from users that validate their trips, can be considerable [33]. Therefore, more advanced hybrid methods have been studied, as have multiple rules and machine-learning specializing in both trajectories and contexts. One hybrid method consists of the following six steps [26]: The first step is trajectory cleansing, based on the accuracy provided by the AGPS; the second step is rule-based detection of stop candidates, where stops are points

within a 50-meter range and a 1-minute time window. The third step checks for stop candidates against users' frequent stop locations. The fourth step merges the resulting stops, with a rule-based algorithm configured with various range and time thresholds. The fifth step detects "still" mode, with a learned classifier based on acceleration. The sixth step removes, after mode detection, any orphan stop left.

3.5. Towards a Standardized Measurement of Performance

All of the aforementioned methods are very critical for the classification steps downstream in the process, and they all lack of flexibility in adapting to different thresholds, which might depend on some users, context, or both. However, the choice of trip segmentation method determines the object to be classified in the next step of the process, which can be a single observation, such as a GPS point, or a set of observations, such as a GPS segment. Consequently, two methods presenting the same classification score might be very different, depending on whether these methods target points or segments. It is very unlikely that the same number of points and segments will identify two analogous trips in terms of space and time. Therefore, comparing the performance score between point- and segment-based methods is misleading. The scores presented in Table 1, 4 and 7 are not comparable, nor harmonized. Since scores and respective results reflect the case of correct classifications related, e.g., to a stage, a trip, an excursion or the whole day, harmonization attempts should take these cases explicitly into account.

Prelipcean et al. [91] introduce penalty systems and metrics that look at where these methods lead to errors, and provide meaning to the comparison among different segmentation techniques. In particular, with respect to the ground truth, if precision and recall identify "hits" and "misses" of a classifier (the broadly used F1-score is the harmonic mean of precision and recall) from such measurements, we do not understand how the error depends on over- or under-segmentation, e.g., of the trajectory that this method classified. Since errors in trajectory segmentation propagate to the classification of the trajectories, and classification performance depends on how the segmentation inference aligns with the ground truth, these penalties are proportional to time and space of segments misaligned with the ground truth. This is in opposition to previous studies where a count of the editing operations was proposed [92]. Interestingly, with this metric, point-based trajectory segmentation techniques seem to outperform segment-based techniques [91]. Since

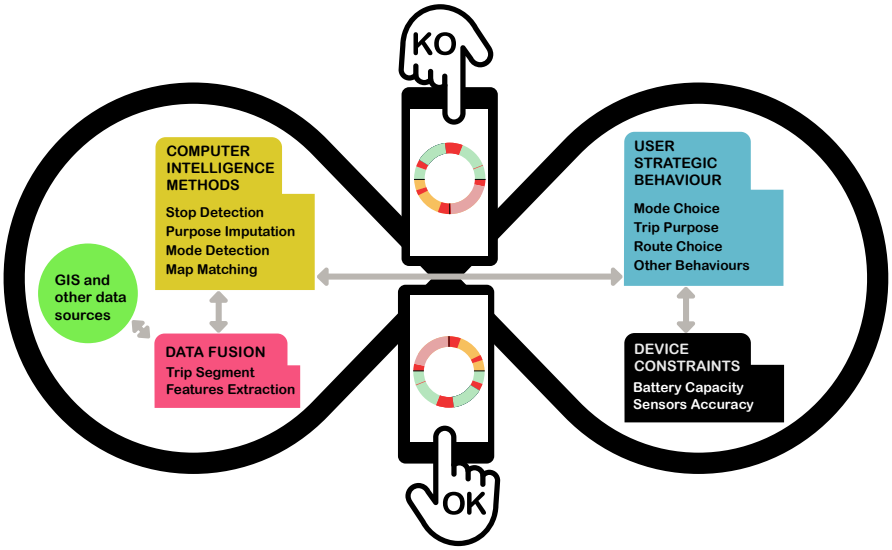

Figure 3: Validation loop in smartphone-based user activity monitoring.

both segment- and point-based classifiers discard any segment below a certain threshold of (e.g.) GPS observations—which in the first case can be two magnitudes higher than in the second case—an intuitive explanation is that segment-based classifiers are incapable of classifying a larger fraction of a dataset.

3.6. Human Activity Recognition in Mobility

To support the modeling of activity and travel choices at the heart, for example, of activity-based models [93], human activity recognition in mobility must include both stop, mode and purpose of any trip. The combination of feature extraction techniques and computer intelligence algorithms allows for a capturing of the correlation between features and the user's strategic choices. As technology evolves, the inference of users' strategic choices in the form of a travel-diary and user validation by means of such a diary (see Fig. 3), enable continuous improvement of the acceptable truth asymptotically approaching the theoretical ground truth. Computer intelligence algorithms are tightly coupled with the data necessary to allow and refine the inferences. Given an initial validated dataset, their performance can be

measured only by comparing inferences with the ground truth (see Fig. 3). Errors propagate from trajectory segmentation, to trajectory classification, and then to the travel-diary generation [91]. Therefore, it is likely that errors propagate to the ground truth. From this standpoint, the output of this process might lead to systematically biased predictions. In SBTS, machine-learning is just a tool used to capture the information represented by data. The quality of models has a strong influence on the quality of the ground truth we can collect through travel-diaries, and vice-versa.

There is consensus in the field about the lack of standardization for validating and comparing competing classifiers. There are several studies where, even though classifications are performed on the same dataset, differences in number and quality of classes predicted and in validation setup are enough to make F1-score comparisons meaningless. For example, F1-scores obtained as average on a 5-label transport mode classification task and a 5-fold cross-validation [89], cannot be compared with F1-scores from a 4-label transport modes classification task, computed on a random test-set only (hold-out method) [90].

We have identified three approached that allow for a comparison to be made between different methods and datasets. The first is the same aforementioned penalization solution to ease the comparison between point- and segment-based classifiers [91]. The second approach could provide a standardized baseline by combining a public dataset and a cross-validation workflow [40]. The dataset includes the observations of 18 sensors on three users made over a period of $2,812$ hours' worth of labeled data. Labels include the position of the phone as: in the hand, at the torso, at the hip, and in a bag. The workflow for cross-validation covers three tasks: user-independent, phone position-independent, and time-invariant. At the end of the three tasks, each one accomplished with manifold cross-validation, the paper suggests the standard deviation of F1-scores computed across users, phone positions, and time periods as the benchmark of the predictive power of a model. This workflow cannot be applied in most of the datasets available, which are not as rich; for example, the widely used Geolife [94] provides GPS trajectories and transport mode labels only (see Table 1). The third approach leverages the Weka software [95], where several machine-learning algorithms are available off-the-shelf. Based on Weka software, Ectors et al. [96] compare a few rule-based and probabilistic machine-learning algorithms for purpose imputation on the same dataset.

However, we found no attempts at combining these three approaches,

which are complementary to comparing different methods, but not self-sufficient. Another step should consider the feature extraction process. Indeed, this process is also subject to attempts of standardization. One candidate method is "minimum redundancy maximum relevance" [40] (MRMR, see Table 3). For classifiers relying on deep learning though, this feature extraction method is not effective, as the neural network extracts the features autonomously. In this case, the new challenge is finding optimal hyper parameters for the neural network. Such hyper parameters may include, for example, architecture configuration, activation functions, batch size, regularization factor, and optimization step. [97] propose an approach to selecting these hyper parameters automatically, moving towards standardized deep learning method optimization. Still, we did not find applications in this field; instead, optimal hyper parameters are still a craftsman product [90, 89, 98].

3.7. Implications for Transport Science

The choice of complementary sensors, such as the gyroscope, could mitigate the challenges that most of the algorithms encounter in discriminating between, for example, bike and walk or bike and bus in congested urban contexts. Similarly, the magnetometer could help distinguish between rails and cars, and the accelerometer between bike and e-bike. However, these high-frequency sensors require online rather than offline classifiers. Offline classifiers would suffer from the large footprint of the data, which would in turn have a negative impact on smartphone users' data plan and battery. This would ultimately lead users to dropout from travel surveys.

Several studies exhibit how useful GIS information can be on mode detection. However, when classifying the complement of the same trajectory, studies on purpose imputation expose the challenges associated with the proximity of heterogeneous points of interest, as various trips can start for different purposes and end in the same spatial range. In such a case generally helpful, personal patterns and a limited amount of personal information proved to support more accurate predictions (see Table 3 against Table 1, and Table 6 against Table 4).

Nevertheless, among the studies identified for map-matching, we find no examples of personal information use (see Table 9). Even in the assumption of unavailability of any personal information, map-matching and consequent route-choice records would amplify the impact of transport mode and trip purpose classification (see Table 7). Expressing a trajectory as a sequence of links and nodes on the transport network, instead of longitude and latitude,

pinpoints specific micropatterns. Furthermore, it potentially reduces the confusion that users often face while validating their travel-diaries in the presence of GPS outliers.

For map-matching, we identify two problems. First, most of the methods specialize in cars and road network for cars, and few or none refer to emerging modes such as e-bikes and e-scooters (see Table 8). Second, in the literature, we did not find a good representation of adequate datasets and ground truth quality levels (see Table 9). In the first case, the assumption that GPS points should belong to the road network does not hold. Map-matching for modes different from cars requires degrees of freedom to allow transit on, for example, sidewalks and bicycle lanes, often not mapped—few studies pinpoint this problem. In contrast, emerging shared modes such as e-bikes and e-scooters imply behaviors not strictly coherent with the mapped network. Furthermore, these emerging modes are introducing new public transport mode-chain-types with irregular patterns, alternating traditional public transport and emerging shared modes. The former offers reliable timetables, while the latter is volatile, as it depends on vehicle availability. Still, [99] show that looking at meaningful mode-chain-types also represent a tool to improve trip classification.

From the direct experience testing Mobile Market Monitor and TRAVELVU on a small user base, we realize that the sample of literature reviewed in this work does not express the differences between a raw trajectory, such as the one that SBTS use to generate travel-diaries, and a processed trajectory, such as the one that SBTS may output as ground truth. The first trajectory presents a level of noise that could even ease trip segmentation process and subsequent classification on uni-modal segments. The lack of noise of in the second trajectory, in contrast, might prevent accurate travel-diary generation. These obvious differences have an impact on the choice of method and performance of any transport-related analysis, such as for mode detection. For example, we expect better generalization of Bayesian temporal models or artificial neural network methods in the first case, and machine-learning techniques such as random forest or support vector machines in the second case.

Further, Tables 3, 6, and 9 clearly show that while artificial neural networks and temporal models do not require particular feature extraction methods, machine-learning approaches such as random forest or support vector machines must rely on time-series feature extraction. Hence, to find the best classification method, e.g. for transport mode, any attempt at ranking

should be considered in light of whether the trajectories of interest embody any pre-processing, and possibly which one. A possible indicator is the proportion of point loss on the dataset after the application of simple filters, e.g. on point speed and time gaps between points.

For travel-diary generation in presence of multiple sensors and large dataset artificial neural networks seem very promising. artificial neural networks are flexible in learning with and without labels. They also act as powerful dimensionality-reduction, information-compression, and feature-extraction to for simultaneous signal processing of multiple sensors monitoring the same event, and signaling at different and irregular frequencies. Let us consider, for example: (i) smartwatches and other bio-metric devices complementary to smartphones [100]; (ii) ongoing software integration between cars and smartphones, which include navigation and INS sensors [101]; and (iii) development of edge-computing to augment the processing power of smartphones when consuming cloud services [102], where users' mobility patterns are studied to reduce service-latency in the information-technology-network.

A holistic approach could amplify the impact of studies sharing the scope of those identified in this review. Smartphones' onboard sensors represent only a fraction of the collectible signals, and the surveyed literature seem not fully aware the quickly-evolving context surrounding smartphone devices. To release new potential towards the disambiguation of transport patterns that in congested urban areas look exactly the same for the surveyed methods, while contrasting the curse of dimensionailty [103], this field requires a new perspective. Compared to the advances in other fields, such as computer vision or social networks, transport science seems only at the beginning of the exploration of artificial neural networks .

4. Discussion

SBTS depends on a sophisticated multi-sided platform which is subject to often conflicting interests over the resources available, beginning with the battery. In current versions, the OS orchestrates the applications' use of sensors and battery, and some OS preclude direct access to AGPS. Therefore, developers have limited configuration possibilities. Furthermore, the data collected through these platforms is affected by large standard deviation, severe errors, and noise due to exogenous elements.

4.1. Sensors

When a smartphone outputs a location signal, whether the location comes from the onboard AGPS, from the triangulation with GSM antennas, the car GPS, or another external GPS connected to the smartphone, developers are not allowed to know. If not properly handled, this uncertainty may negatively affect datasets, method classification performance, user validation and finally ground truth.

Smartphone onboard sensors represent only a fraction of the bio-metric and ambient sensors that could be connected with these devices. [100] present a survey of activity classification from wearable sensors. Differing effective frequencies of each sensor, e.g., $1-10\ Hz$ for GPS, or $> 20\ Hz$ for accelerometer, require flexible frameworks as for joint features extraction, compression, and analysis. From this standpoint, artificial neural networks seem to have potential.

4.2. Data Sources

From the perspective of smartphone-related trajectories, a better understanding of travel behavior requires the standardization of measures relevant for travel patterns, which should also rely on standard datasets. The options available are a good starting point, but still seem insufficient. For example, let us consider the following datasets. (i) [104] deliver real GPS trajectories collected in the USA from real smartphones, in which ground truth, available on trip mode and not trip purpose, is generated synthetically to protect privacy exposure (users follow instructions provided by a custom App). (ii) [40] offer trajectories collected in the UK from multiple smartphone sensors at relevant frequencies, and from smartphones of the same model positioned on various part of the body, providing ground truth for trip mode only. (iii) [94] include GPS trajectories from China, with ground truth on trip mode for 69 users out of 189. (iv) [105] supply GPS trajectories collected in various parts of the world for map-matching, but not multi-modal. (v) [106] propose onboard high-frequency sensors with ground truth on transport mode, collected in Italy from multiple smartphones and users, but where GPS is unavailable. (vi) [107] provide data from over 72 wearable sensors, collected indoors with ground truth on performed activities, and no GPS. (vii) [108] offer data collected in Switzerland over 18 months from 185 users of the Nokia N95 device with multiple sensors, including, for example, AGPS, accelerometer, Bluetooth, trip purpose labels, and no transport modes.

4.3. Methods

The collection of any acceptable ground truth depends on the reliability and accuracy of underlying measurement methods. The vast choice of alternatives requires a standardized way of comparing competing methods. Existing literature offers effective penalization systems for classic performance scores [22]. Invitations on standardized mode detection are available in form of feature extraction and cross-validation workflows [40]. However, these attempts do not seem sufficient to cover mode detection, purpose imputation, and map-matching at the same time across existing and emerging methodologies.

We identified excellent alternatives. Some perform best on low-resolution trajectories. Other classifiers are tight (e.g.) to the location where GPS trajectories are fused with data from GIS, users' personal information, or both. Among the best performers in terms of accuracy measurement, in general, we find: support vector machines, fuzzy logic, random forests, and probabilistic models (e.g., Hidden Markov Models). Classic rule-based algorithms might not perform at the same accuracy level. However, they are still competitive when the application scenario is stable, and if execution speed and scalability are a priority over accuracy.

Methods based on artificial neural networks are rising quickly and are applicable across mode detection, purpose imputation, and map-matching, as probabilistic and Bayesian methods unlike other machine-learning techniques. For map-matching and purpose imputation, for example, we find applications combining GPS and GIS, while for stop and mode detection, we find applications with GPS only. Particular configurations of these methods, such as variational auto encoders and deep kalman filters, which represent the convergence with Bayesian methods, could offer a background facilitating methodological convergence that might also allow for a breakthrough in this mature field of research.

4.4. Ground Truth

Whether a study targets, for instance, the whole day, week, month, season or year, modelers need a correct dataset ideally of a whole period. If this is not the case, the value of the whole dataset is limited. Since a "person to device" validation might introduce further errors; their magnitude and their impact on machine-learning methods performance should be investigated. We find no attempt of self-learning on multi-sensor datasets, which would raise expectations on a "device-to-device" ground truth evolution. We could

achieve full automation of both travel-diary generation and validation by using independent measurements of the same event to substitute traditional labels with pseudo-labels. For example, instead of learning from labels, artificial neural networks could learn GPS patters to reconstruct accelerometer patterns, and vice-versa. Meanwhile, where machine-learning algorithms do not provide correct travel-diaries to the user, "person to device" interaction could be enhanced by introducing the possibility for the user: (i) to trigger a specialized automatic evaluation of such segments; and (ii) to flag whether he or she was unable to correct the mistakes (see Fig. 3).

5. Conclusion

In transport science, the process of methodological perfection between paper-and-pencil personal interviews, and computer assisted personal interviews [109], towards computer assisted telephone interviews [110], and computer assisted web interviews [111] is still evolving towards SBTS [25, 112]. The leap between paper and computer determined a structural impact on the surveying costs, requiring software, IT-infrastructure, and personnel-training. According to [113], the shift to computer assisted web interviews requires to fall back to telephone interviews in cases where the web interviews are incomplete.

From computer to smartphones, the impact seems negligible both on software and IT-infrastructure costs. In contrast, the impact on human resources seems to determine a significant reduction of personnel, and a shift towards highly specialized and more expensive skills of data scientists required to deploy a SBTS. Consequently, under a certain volume-threshold of, e.g., surveyed users in time, traditional surveys could be still competitive in terms of cost. However, to push transport science boundaries under the constraint of Big Data—which traditional travel surveys are unable to satisfy—SBTS bring a huge scalability potential and support higher resolution datasets, handling users during time horizons longer than just one day.

To expose SBTS potential, this paper selects and summarizes information on SBTS relevant for a qualitative comparison of the methods focusing on mode detection, purpose imputation, and map-matching. To ease such a comparison, since the standardization process in the field is still ongoing, we organized the literature into tables, which include information about classification objectives, datasets employed in the experiments, and validation approach of both data and experiments. Besides, by listing sensors, features,

and dataset that each of the related works depends on, we identify the main methods underlying the process of ground truth generation.

Comparison based only on scores reflecting different variables, such as accuracy and F-score, is misleading. As we find, scores depend on the underlying dataset, trajectory segmentation, classification method and experiment design. Evaluation of larger segment units leads to discarding significant portions of a dataset. The classification task is relatively more difficult with a larger number of classes. The accuracy bias is relatively lower when performing cross-validation, and when processing more representative datasets. For example, Tables 1 and 2 for mode detection, Tables 4 and 5 for purpose imputation, as well as Tab 7 and 8 for map-matching expose, from another perspective than Prelipcean et al. [91], that methods performance is beyond dry scores. When comparing methods, newcomers in this field would certainly benefit from considering task complexity, representativeness of the supporting dataset, and validation method. For example, task and method complexity, features collection and extraction cost (see Tables 3, 6, and 9).

A converging thrust in the field seems represented by simultaneous methods focusing on, e.g., mode detection to improve map-matching or purpose imputation, and vice-versa. To support the disambiguation of travel patterns that are still challenging to detect in congested urban areas, for the future, emerging applications of artificial neural networks seem to support further fruitful convergence. The study of smartphones onboard sensors in addition to other streams collectible through smartphones—from GIS, wearable sensors, or edge-computing—would benefit from the artificial neural networks flexible framework. This technology can be exploited on the one hand to learn from large and heterogeneous data streams, and on the other hand to compress and store such BIG bulk of information through relatively few trained parameters. To support the standardization of relevant measures for transport behavior, efforts should also be directed towards the solution of privacy concerns that represent an obstacle, in this field, for the generation of open-access datasets.

6. Abbreviations

AGPS: Assisted Global Positioning Systems
CPU: Central Processing Unit
GIS: Geographic Information Systems
GPS: Global Positioning Systems

GPU: Graphical Processing Unit
INS: Inertial Navigation Systems
OS: Operation Systems
P2D: Person-to-Device
P2P: Person-to-Person
SBTS: Smartphone-based Travel Surveys

Table 1: Classification task ranked by difficulty and score, for mode detection

Ref.		No.	Classes	Score	Metric	Validation	Area
1	[88]	6	Walk, Bike, Bus, Car, Rail, Plain	86.5%	Accuracy	Hold-out	Bieijing
2	[114]	6	Car, Train, Bus-Tram-Metro, Foot, Bicycle, Other	70,00%	Accuracy	n.p.	Netherlands
3	[115]	5	Walk, Bike, Bus, Car, Rail	96.8%	Accuracy	Manifold-cross-validation	Minnesota
4	[116]	5	Walk, Bike, Bus, Car, Run	95.1%	F-Score	Manifold-cross-validation, Out-of-bag-estimate	Tennessee
5	[11]	5	Walk, Bike, Bus, Car, Rail	94,00%	Accuracy	Manifold-cross-validation	Leuven
6	[117]	5	Walk, Bike, Run, in-Vehicle, Stationary	93.8%	Accuracy	Hold-out	Georgia (USA)
7	[118]	5	Walk, Bike, Bus, Car, Rail	93.45%	F1-Score	Manifold-cross-validation	Bieijing
8	[119]	5	Walk, Bike, el-Bike, Car, Bus	92.74%	Accuracy	Manifold-cross-validation	Shanghai
9	[51]	5	Walk, Bike, Car, Bus, Rail	92.4%	Accuracy	n.p.	Copenhagen
10	[120]	5	Walk, Bike, Public transit, Car, Car and Public transit	88,00%	F1-Score weighted average	Manifold-cross-validation	Montreal

Continued on next page

Table 1: Classification task ranked by difficulty and score, for mode detection

Ref.	No.	Classes	Score	Metric	Validation	Area
11 [89]	5	Walk, Bike, Bus, Car, Rail	84.8%	F-Score	Manifold-cross-validation	Bieijing
12 [56]	5	Auto, Bus, Streetcar, Bike, Walk	82,00%	F1-Score weighted average	Hold-out	Toronto
13 [7]	5	Walk, Bike, Bus, Car, Rail	82,00%	Accuracy	n.p.	Netherlands, (Geurs et al., 2015)
14 [121]	5	Walk, Bike, Bus, Drive, Train	76.4%	F1-Score weighted average	Manifold-cross-validation	Bieijing
15 [90]	4	Walk, Bike, Bus, Car	98,00%	Accuracy	Hold-out	Bieijing
16 [122]	4	Walk, Bike, Bus, Car	94.7%	Accuracy	Hold-out	New-Zeland
17 [32]	4	Walk, Bike, Transit, Car	91.8%	Accuracy	Manifold-cross-validation	Montreal

Continued on next page

Table 1: Classification task ranked by difficulty and score, for mode detection

Ref.	No.	Classes	Score	Metric	Validation	Area
18 [123]	4	Walk, Bike, Bus, Car	90.7%	F1-Score	Manifold-cross-validation, Out-of-bag-estimate	Beijing. 1 week BUS trajectories, 1000 trajectories from Open Street Map (OSM)
19 [124]	4	Walk, Bike, Transit, Car	83.4%	Accuracy	Manifold-cross-validation	Montreal

Continued on next page

Table 2: Dataset ranked by number of users, for mode detection

Ref.	Person-day	Users	Ground Truth	Observations	Time	Area	Smartphone App
5 [11]	24,900	8,303	Validated-by-respondents	30,000 trips 3,960,243 GPS points 340,000 km	n.p.	Leuven	Routecoach
19 [124]	88,630	6,846	Validated-by-respondents (102,904 trips)	623,718 trips	2 months col-leciton period	Montreal	MTL Traject App
17 [32]	88,630	6,846	Validated-by-respondents (P2D)	102,904 trips	2 months col-leciton period	Montreal	MTL Traject App
10 [120]	88,630	6,846	Validated-by-respondents (P2D)	131,777 trips 33 mln GPS points	2 months col-leciton period	Montreal	MTL Traject App

Continued on next page

Table 2: Dataset ranked by number of users, for mode detection

Ref.		Person-day	Users	Ground Truth	Observations	Time	Area	Smartphone App
2	[114]	40,208	1,104	Validated-by-respondents (P2D)	n.p.	7,395 days	Netherlands	GPS logger and Web based validation
13	[7]	n.p.	600	Validated-by-respondents	60,000 trips	3 batches per 1 month each	Netherlands, (Geurs et al., 2015)	Move Smarter
8	[119]	1,248	202	Validated-by-respondents	4,685 Trip-legs	n.p.	Shanghai	Shangai City - Smartphone Based Travel Survey
14	[121]	4,000	189	Partially Validated-by-respondents (69 respondents)	17,621 trajectories 1,292,951 km 50,176 hours	3 years collection period	Bieijing	Geolife (Zheng and Fu, 2011)
9	[51]	644	101	Validated-by-respondents (P2P)	6,419,441 GPS points 1,783 h of travel	3-5 days per respondent	Copenhagen	GPS logger

Continued on next page

Table 2: Dataset ranked by number of users, for mode detection

Ref.	Person-day	Users	Ground Truth	Observations	Time	Area	Smartphone App
16 [122]	372	76	Validated-by-respondents	760,000 GPS observations, 530 hours trajectories	2 months per respondent	New-Zeland	Advanced Travel Logging Application for Smartphones II (ATLAS II)

Continued on next page

Table 2: Dataset ranked by number of users, for mode detection

Ref.	Person-day	Users	Ground Truth	Observations	Time	Area	Smartphone App
18 [123]	4,000	> 69	Validated-by-respondents	n.p.	n.p.	Beijing. 1 week BUS trajectories, 1000 trajectories from Open Street Map (OSM)	Geolife (Zheng and Fu, 2011), Journeys API [1], OpenStreetMap [2]
11 [89]	4,000	69	Validated-by-respondents	n.p.	3 years collection period	Bieijing	Geolife (Zheng and Fu, 2011)

Continued on next page

[1]Journeys API, retrieved from web 01/01/2019, http://wiki.itsfactory.fi/index.php/Journeys_API
[2]Open-source Trajectories , retrieved from web 01/01/2019, https://www.openstreetmap.org/traces

Table 2: Dataset ranked by number of users, for mode detection

Ref.	Person-day	Users	Ground Truth	Observations	Time	Area	Smartphone App
15 [90]	4,000	69	Validated-by-respondents	n.p.	3 years collection period	Bieijing	Geolife (Zheng and Fu, 2011)
1 [88]	4,000	69	Validated-by-respondents	n.p.	3 years collection period	Bieijing	Geolife (Zheng and Fu, 2011)
7 [118]	4,000	69	Validated-by-respondents	n.p.	3 years collection period	Bieijing	Geolife (Zheng and Fu, 2011)
6 [117]	n.p.	12	Validated-by-respondents	n.p.	6 days per respondent	Georgia (USA)	Self Developed App
3 [115]	n.p.	6	Validated-by-respondents	347,719 GPS points in 96.59 h (1Hz) 1.7 mln points Acceleration in 98.62 h (5Hz)	n.p.	Minnesota	Self Developed App

Continued on next page

Table 2: Dataset ranked by number of users, for mode detection

Ref.	Person-day	Users	Ground Truth	Observations	Time	Area	Smartphone App
12 [56]	n.p.	n.p.	n.p.	n.p.	50 hours	Toronto	Self Developed App
4 [116]	n.p.	n.p.	Validated-by-respondents	n.p.	n.p.	Tennessee	Self Developed App

Continued on next page

Table 3: Methodlogy and features, for mode detection

Ref.	Method	Main Features	AGPS	INS	GIS
16 [122]	Nested Logit Model, Muiltinomial Logistic Regression, Multiple Discriminant Analysis	Skewness of speed distribution, Share of travel time with speed (m/s) $\in [2, 8)$, Share of travel time with speed (m/s) $\in [8, 15)$, Maximum speed, 95% percentile acceleration, Maximum acceleration, Acceleration variance, Direct distance $origin \rightarrow destination$, Travelled distance $origin \rightarrow destination$	yes	no	no
2 [114]	Rule-based	Distance $GPS \rightarrow$ Points-of-interest, Distance $GPS \rightarrow LandUse$	yes	no	yes
12 [56]	Neural Network	Speed, Acceleration, Magnetic field, Satellites number	GPS	Accelerometer Magnetometer	no
11 [89]	Convolutional Neural Network, Random Forest, Key Nearest Neighbor, Support Vector Machines, Multy Layer Perceptron	Speed, Acceleration, Jerk, BearingRate	yes	no	no
14 [121]	SEmi-Supervised Convolutional Autoencoder	GPS Points: Relative Distance, Speed,	yes	no	no

Continued on next page

Table 3: Methodlogy and features, for mode detection

Ref.	Method	Main Features	AGPS	INS	GIS
4 [116]	Random Forest, Bagging Model, Support Vector Machines, Key Nearest Neighbor, Max-Dependency Min-Redundancy	Acceleration spectral entropy, Acceleration range, Max angular velocity, Average absolute acceleration, Average angular velocity	yes	Accelerometer, Gyroscope, Rotation Vector	no
15 [90]	Recurrent Neural Network, Hampel filter	Speed, Average speed, Standard deviation speed	yes	no	no
18 [123]	Bayesian Classifier, Neural Network, Random Forest, Auto Encoder	Maximum acceleration, Maximum speed, Minimum acceleration, Minimum Speed, Average acceleration, Average speed, Acceleration variance, Speed variance, Speed skewness, Speed kurtosis, Acceleration Skewness, Acceleration Kurtosis	yes	no	no
3 [115]	Random Forest, Key Nearest Neighbor, Principal Component Analysis, Recursive Feature Elimination	Average change in acceleration ($\Delta T = 120s$), 80% percentile speed ($\Delta T = 120s$), Variance change in acceleration ($\Delta T = 120s$), Maximum speed ($\Delta T = 120s$), Average speed ($\Delta T = 120s$), Average change in speed ($\Delta T = 120s$)	yes	Accelerometer	no
9 [51]	Fuzzy Logic	95% percentile acceleration, 95% percentile speed, Median speed, Network segment	GPS	no	yes

Continued on next page

Table 3: Methodlogy and features, for mode detection

Ref.	Method	Main Features	AGPS	INS	GIS
5 [11]	Support Vector Machines	Distance From (DF) motorway, DF railway, DF bicycle lane, DF bus stop, DF railways station, DF car parking, DF bicycle parking, DF bus line	yes	no	yes
13 [7]	Bayesian Classifier	Personal trip history, Speed, Altitude, Longitude, Latitude, Public transport time-table	yes	Accelerometer	yes
8 [119]	Bayesian Network	Average speed, 95% percentile speed, Average absolute acceleration, Travel distance, Average heading change, Low-speed-rate (as the ratio of points with speed¡threshold)	yes	no	no
17 [32]	Counvolutional Neural Network augmented with ensemble method, with Random Forest as meta learner	GPS Points: Relative Distance, Speed	yes	no	no

Continued on next page

Table 3: Methodlogy and features, for mode detection

Ref.	Method	Main Features	AGPS	INS	GIS
10 [120]	Random Forest	Measures Between Origin-Destination: Cumulative and Direct distance (m), Travel Time (min.), Average and 85th percentile speed (km/h), Maximum, Minimum Difference between Min. and Max. Acceleration (km/h^2), Minimum and Maximum slope; Max time interval (min) and Max distance (m) between each consecutive pair of GPS point; Time of day and Time of Week; Age, Gender, Occupation; Average value of residential buildings around each individual's home (in 250 meters radius); Direct Distance between the origin and nearest public transit stop; Direct Distance between the destination and nearest public transit stop; Average value of residential buildings around each individual's home (in 250 m radius)	yes	no	yes
19 [124]	Semi-supervised Generative Adversarial Networks	GPS Points: Relative Distance, Speed	yes	no	no

Continued on next page

Table 3: Methodlogy and features, for mode detection

Ref.		Method	Main Features	AGPS	INS	GIS
6	[117]	Random Forest with 3 layers	Speed, *Acceleration* − *Gravity*, Fast Fourier Transform (Frequency Domain), Energy of the signals, Sum of spectral coefficients	yes	Accelerometer	no
1	[88]	Random Forest	85% percentile speed, Average speed, Median speed, Medium velocity rate, High velocity rate, Low velocity rate, Travel distance	yes	no	yes
7	[118]	Auto Encoder, Deep Neural Network	Average speed, Travel distance, Average acceleration, Head direction change, Bus stop closeness, Subway line closeness	yes	no	yes

Continued on next page

Table 4: Classification task ranked by difficulty and score, for purpose imputation

Ref.	No.	Classes	Score	Metric	Validation
20 [59]	15	Work, Study, Shopping, Social Visit, Recreation, Home, Business Meeting, Change mode/Transfer, Pick-up, Drop-off, Meal/Eating break, Personal Errand/Task, Medical/Dental, Entertainment, Sport/Exercise	98.68%	F1-Score	Out-of-bag-estimate
21 [125]	10	Study, Social Visit, Recreation, Home, Service, Paid Work, Daily Shopping, Non-daily Shopping, Help parents/cildren, Voluntary work	96.8%	Accuracy	Out-of-bag-estimate
22 [50]	9	Work, Shop, Service, Recreation, Home, Pick-up, Drop-off, Business Meeting, Other	79.8%	Accuracy	Out-of-bag-estimate
23 [12]	8	Work, Study, Shop, Social Visit, Home, Eeating Out, Pick-up, Drop-off	96.53%	Accuracy	Hold-out
2 [114]	7	Work, Study, Shop, Social Visit, Recreation, Home, Other	43%	Accuracy	n.p.
10 [120]	6	Education, Health, Leisure, Shopping/Errands, Home, Work	72%	F1-Score weighted average	Manifold-cross-validation

Continued on next page

Table 5: Dataset ranked by number of users, for purpose imputation

Ref.		Person-Users	day	Ground truth	Observations	Time	Area	Smartphone App
10	[120]	88,629	6,845	Validated-by-respondents (P2D)	131,777 trips, 33 mln GPS points	1 month colleciton period	Montreal	MTL Trajject App
2	[114]	40,208	1,104	Validated-by-respondents	n.p.	7,395 days	Netherlands	GPS logger and Web based validation
20	[59]	7,856	793	Validated-by-respondents (P2D)	22,170 days, 130 mln GPS points	5-14 days per respondent	Singapore	Futur Mobility Survey
21	[125]	n.p.	329	Validated-by-respondents (P2D)	10,545 activities	3 month per respondent	Netherlands (Rotterdam)	GPS logger and Web based validation

Continued on next page

Table 5: Dataset ranked by number of users, for purpose imputation

Ref.	Person-day	Users	Ground truth	Observations	Time	Area	Smartphone App
23 [12]	2,409	321	Validated-by-respondents (P2P)	7,039 trips	7-12 days per respondent	Shanghai	Shangai City - Smartphone Based Travel Survey
22 [50]	n.p.	156	Validated-by-respondents	6,938 activities	7 days	Zurich	Self Developed App

Continued on next page

Table 6: Methodlogy and features, for purpose imputation

Ref.	Method	Main Features	AGPS	INS	GIS
2 [114]	Rule-based	Distance $GPS \rightarrow$ Points-of-interest, Distance $GPS \rightarrow LandUse$	GPS	no	yes
21 [125]	Random Forest	Activity Duration, Activity Start Time, Travel Time to Activity, Distance $GPS \rightarrow$ Points-of-interest	GPS	no	yes
20 [59]	Bagging Decision Tree, Random Forest	Activity Probability, Distance-based Empirical Probability, Activity Transition Probability, Activity Duration	yes	Accelerometer	yes
22 [50]	Clustering, Random Forest	Start Time, End Time, GPS points density, Age, Education, Income, Mobility Ownership, Activity Duration, Walk Percentage	yes	Accelerometer	yes
23 [12]	Multy Layer Perceptron, Particle Swarm Optimisation, Multinomial Logit, Support Vector Machines, Bayesian Network	Age, Gender, Education, Working Hours, Income, Time of Week, Activity Duration, Time of Day, Transportation Mode, Distance $GPS \rightarrow$ Points-of-interest, Distance $GPS \rightarrow LandUse$	yes	no	yes

Continued on next page

Table 6: Methodlogy and features, for purpose imputation

Ref.	Method	Main Features	AGPS	INS	GIS
10 [120]	Random Forest	Features returned by Open Trip Planner [3] itinerary: GPS tracks average speed, Time interval between the first and last GPS track of a trip, Average distance between consecutive GPS point, Attributes from, Itinerary length, Total transit time of each returned, Total walking time of each itinerary, Total waiting time of each itinerary, Total travel time, Number of transfers, Walking distance, Itinerary average speed Attributes from GPS Tracks, Difference between GPS tracks length and itinerary length, Overlapping percentage of itinerary and GPS tracks	yes	no	yes

Continued on next page

[3]Open Trip Planner (OTP) retrieved from web 01/01/2019, https://github.com/opentripplanner/OpenTripPlanner)

Table 7: Map-matching task ranked by difficulty and score

Ref.	Mode	Category	Score	Metric	Validation
24 [17]	Walk, Bike, Car, Metro	Multimodal, Global, Shortest-path	[80%, 99%]	Path Similarity Indicator	n.p.
25 [126]	Bicycle	Match when possible, build when needed	n.p.	n.p.	n.p.
26 [127]	Car	Unimodal, Incremental, Point-based	99.2%	$A = \frac{\#(correctlymatchedGPSpoints)}{\#(TotalGPSpoints)}$	n.p.
27 [128]	Car	Unimodal, Incremental, Point-based	99.8% (sub-urban), 97.8% (urban)	$A = \frac{\#(correctlymatchedGPSpoints)}{\#(TotalGPSpoints)}$	n.p.
28 [129]	Car	Unimodal, Incremental, Shortest-path	98%	Accuracy	n.p.
29 [55]	n.p.	Unimodal, Global, Shortest-path	[80%, 99%]	Path Similarity Indicator	n.p.
30 [130]	Taxi	Unimodal, Incremental, Point-based	93.58%	Prediction Accuracy of next road by the road having the maximum probability	Hold-out
31 [131]	Taxi	Unimodal, Incremental, Shortest-path, Supervised, Unsupervised	100% (1s resolution), > 90% (30s resolution)	Accuracy	Manifold-cross-validation

Continued on next page

Table 7: Map-matching task ranked by difficulty and score

Ref.	Mode	Category	Score	Metric	Validation
32 [132]	Taxi	Unimodal, Incremental, Point-based	87.18%	$A = \frac{\#(correctlymatchedGPSpoints)}{\#(TotalGPSpoints)}$	Hold-out
33 [18]	Dataset 1: Taxi. Dataset 2: n.p.	Unimodal, Global, Shortest-path	91.3%	Average F-Score with: $Precision = \frac{Length_{correct}}{Length_{matched}}$, $Recall = \frac{Length_{correct}}{Length_{truth}}$, Input-to-output latency (Timelines)	Hold-out
34 [133]	Car	Unimodal, Incremental, Point-based	100% (1s resolution), > 90% (30s resolution)	$Accuracy = 1 - E_L$, where $E_L = \frac{(d_- + d_+)}{(d_0)}$, d_- = erroneous subtracted length, d_+ = erroneous added length, d_0 = length of correct route	Hold-out
35 [19]	n.p.	Unimodal, Global, Shortest-path	$A_N > 81\%$, $A_L > 87\%$	$A_N = \frac{\#(correctlymatchedroadsegments)}{\#(allroadsegmentsofthetrajectory)}$, $A_L = \frac{(\Sigma lengthofmatchedroadsegments)}{(lengthofthetrajectory)}$	Hold-out

Continued on next page

Table 8: Dataset ranked by number of links and users, for Map-matching

Ref.	Links	Users	Ground Truth (GT)	Observations	Area	Device
26 [127]	4,605	n.p.	24-channel dual-frequency geodetic receiver	4h Trajectories, 1s resolution	London, sub-urban areas	GPS logger, Gyroscope, Odometer
32 [132]	583	12,000	Hand match supported by Rule Based Algorithm	Training-set: 8,678 GPS points (traces + syntetic from GIS), Test-set: 1,334 GPS points (traces only), 10s resolution	Beijing, urban areas	GPS logger
30 [130]	n.p.	442 + 13,650	No GT available. Hidden Markov Models map-matching results as benchmark with (Newson and Krumm, 2009)	859,195 Traces, 3,709,666 Traces	Porto, Shangai	GPS logger
35 [19]	n.p.	189	Validated-by-respondents (69 users only)	Dataset 1: Syntetic generated from road network (error normally distributed 20 stdev, 0 mean). Dataset 2: 28 GPS Traces (Trips)	Beijing	Geolife, (Zheng et al., 2009)
24 [17]	n.p.	180	No GT available. Unimodal map-matching result as benchmark	10s resolution	Lausanne (CH) Urban and outskirt areas	Nokia EPFL Lausanne (Kiukkonen etal. 2010)

Continued on next page

Table 8: Dataset ranked by number of links and users, for Map-matching

Ref.	Links	Users	Ground Truth (GT)	Observations	Area	Device
31 [131]	n.p.	Dataset 1: 10. Dataset 2: 600	Dataset 1: 1s resolution GPS considered as High Accuracy GT. Dataset 2: no GT	Dataset 1: 700,000 GPS points, 1s resolution. Dataset 2: 600,000 points, 1min resolution	S. Francisco	Mobile Millennium system - GPS logger
33 [18]	n.p.	Dataset 1: 21,807 GPS points, 20 trips, 421 km. Dataset 2: 1,000 trips, 13,139 km	Dataset 1: Manual Check on Map-matched GPS points from higher accuracy source (smartphone), leveraging on knowledge of taxi route. Dataset 2: User validation	Dataset 1: 21,807 GPS points, 20 trips (TAXI), 421 km, 1s resolution. Dataset 2: 13,139 km, 1000 trips. Dataset 3: Syntetic Dataset adding noise to Dataset 1	Singapore	Dataset 1: Custom Smartphone App (Android), Dataset 2: Commercial Smartphone App
34 [133]	n.p.	1	Route planned before data collection and hand match	7,531 GPS points, 80 km, 1s resolution, degraded data simulation	Seattle	GPS logger

Continued on next page

Table 8: Dataset ranked by number of links and users, for Map-matching

Ref.	Links	Users	Ground Truth (GT)	Observations	Area	Device
29 [55]	n.p.	Dataset 1: 1 users. Dataset 2: 3 users.	Dataset 1: Known true path. Dataset 2: no ground truth. Dataset 3: High accuracy GPS device	Dataset 1: 10 points Dataset 2: 25 trips 1041 GPS points, 10s resolution	Lausanne (CH), Urban and outskirt	Nokia EPFL Lausanne (Kiukkonen, Blom, Dousse, Gatica-perez, and Laurila, 2010)
27 [128]	n.p.	n.p.	Tightly-coupled carrier phase GPS receivers integrated with a high-grade inertial navigation system	3363 epochs (sub-urban), 2399 epochs (urban), resolution: 1 epoch/second	Nottingham rural sub-urban, Central London	GPS logger, Digital Elevation Model
25 [126]	n.p.	n.p.	n.p.	128 GPS Traces, 185,000 GPS points, 360 km, 1,088 min	Minneapolis (Twin Cities)	Cyclopath Android App
28 [129]	n.p.	n.p.	n.p.	14,436 GPS points (SIGSPATIAL Cup 2012 DS), 19,080 GPS points, 1s resolution	Seattle Shanghai	

Continued on next page

Table 9: Methodology and main features, for Map-matching

Ref.	Method	Main Features	AGPS	INS	GIS
26 [127]	Fuzzy Logic, Extended Kalaman Filter	Speed, Heading Error, Perpendicular Distance, Horizontal Dilution of Precision	12-channel single frequency high sensitivity GPS receiver	Dead-Reckoning	yes
27 [128]	Rule Based, Extended Kalaman Filter, Integrity check	Altitude, Longitude, Latitude, Traffic flow directions, Road curvature, Grade separation, Travel distance, Heading	GPS	Dead-Reckoning	yes
29 [55]	Probabilistic	Timestamp, Longitude, Latitude, Speed, Heading, Horizontal error Std. Dev., Network error Std. Dev.	yes	no	yes
32 [132]	Feed Forward Neural Network	Longitude, Latitude, Timestamp, Heading	GPS	no	yes
35 [19]	Mixed Method: Topological, Geometric, Probabilistic	Distance GPS(t) $\rightarrow$ GPS(t+1), Distance GPS $\rightarrow$ Network, Shortest path between candidate points on Network, Average speed	yes	no	yes
25 [126]	Hidden Markov Model, Viterbi	Distance GPS $\rightarrow$ Node, Maximum out-degree of the transportation graph	yes	no	Cyclo-path map

Continued on next page

Table 9: Methodology and main features, for Map-matching

Ref.	Method	Main Features	AGPS	INS	GIS
28 [129]	Global Max-weight, Hidden Markov Model, Viterbi	Fréchet distance, Shortest-path	GPS	no	Open Street Map
24 [17]	Probabilistic	Transport mode, Distance, Speed, Acceleration	yes	Accelerometer, Bluetooth Low Energy	yes
30 [130]	Recurrent Neural Network, Long Short Term Memory	Longitude, Latitude, Timestamp, Destination	GPS	no	Open Street Map
34 [133]	Hidden Markov Model, Viterbi	Distance GPS(t) → GPS(t+1), Distance GPS(t) → network (only in range ¡200m)	yes	no	yes
31 [131]	Undirected graph Bayesian Network, Viterbi	Path length, Distance Point projection → GPS, Number of signals, Number of turns, Average speed, Max/min num. Lanes	GPS	no	560,000 links map

Continued on next page

Table 9: Methodology and main features, for Map-matching

Ref.	Method	Main Features	AGPS	INS	GIS
33 [18]	DS 1: Hidden Markov Model, Viterbi, Conditional Random Fields (CRF). DS 2: Multinomial Logit Model, k-shortest path with link-penalty approach.	Path Choice: Free-flow travel time (seconds), Number of traffic signals, Average road class, Number of class changes	AGPS, with WiFi and GPS off	no	yes

Continued on next page

7. Competing interests

The authors declare that they have no competing interests.

References

[1] L. Gong, T. Morikawa, T. Yamamoto, H. Sato, Deriving Personal Trip Data from GPS Data: A Literature Review on the Existing Methodologies, Procedia - Social and Behavioral Sciences 138 (2014) 557–565. URL: http://linkinghub.elsevier.com/retrieve/pii/S1877042814041597. doi:10.1016/j.sbspro.2014.07.239. arXiv:arXiv:1105.4823v1.

[2] M. Ben-Akiva, S. R. Lerman, Discrete Choice Analysis: Theory and Application to Travel Demand, The MIT Press, Cambridge, MA, USA, 1985.

[3] D. Houston, T. T. Luong, M. G. Boarnet, Tracking daily travel; Assessing discrepancies between GPS-derived and self-reported travel patterns, Transportation Research Part C: Emerging Technologies 48 (2014) 97–108. doi:10.1016/j.trc.2014.08.013.

[4] M. G. Karlaftis, E. I. Vlahogianni, Statistical methods versus neural networks in transportation research: Differences, similarities and some insights, Transportation Research Part C: Emerging Technologies 19 (2011) 387–399. URL: http://dx.doi.org/10.1016/j.trc.2010.10.004. doi:10.1016/j.trc.2010.10.004.

[5] C. Renso, M. Baglioni, J. A. F. de Macedo, R. Trasarti, M. Wachowicz, How you move reveals who you are: Understanding human behavior by analyzing trajectory data, Knowledge and Information Systems 37 (2013) 331–362. doi:10.1007/s10115-012-0511-z.

[6] O. Yurur, C. H. Liu, Z. Sheng, V. C. M. Leung, W. Moreno, K. K. Leung, Context-awareness for mobile sensing: A survey and future directions, IEEE Communications Surveys and Tutorials 18 (2016) 68–93. doi:10.1109/COMST.2014.2381246.

[7] T. Thomas, K. T. Geurs, J. Koolwaaij, M. Bijlsma, Automatic Trip Detection with the Dutch Mobile Mobility Panel : Towards Reliable Multiple-Week Trip Registration for Large Samples, Journal of

Urban Technology 0 (2018) 1–19. URL: https://doi.org/10.1080/10630732.2018.1471874. doi:10.1080/10630732.2018.1471874.

[8] J. Gadziński, Perspectives of the use of smartphones in travel behaviour studies: Findings from a literature review and a pilot study, Transportation Research Part C: Emerging Technologies 88 (2018) 74–86. doi:10.1016/j.trc.2018.01.011.

[9] B. V. Wee, D. Banister, How to Write a Literature Review Paper?, Transport Reviews 36 (2016) 278–288. URL: http://dx.doi.org/10.1080/01441647.2015.1065456. doi:10.1080/01441647.2015.1065456. arXiv:arXiv:1011.1669v3.

[10] H. Christiansen, M.-L. Warnecke, The danish national travel survey - declaration of variables tu 2006-17, version 1, 2018.

[11] I. Semanjski, S. Gautama, R. Ahas, F. Witlox, Spatial context mining approach for transport mode recognition from mobile sensed big data, Computers, Environment and Urban Systems 66 (2017) 38–52. doi:10.1016/j.compenvurbsys.2017.07.004.

[12] G. Xiao, Z. Juan, C. Zhang, Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization, Transportation Research Part C: Emerging Technologies 71 (2016) 447–463. URL: http://dx.doi.org/10.1016/j.trc.2016.08.008. doi:10.1016/j.trc.2016.08.008.

[13] J. Auld, C. Williams, A. Mohammadian, P. Nelson, An automated GPS-based prompted recall survey with learning algorithms, Transportation Letters (2009). doi:10.3328/TL.2009.01.01.59-79.

[14] L. Shen, P. R. Stopher, Review of GPS Travel Survey and GPS Data-Processing Methods, 2014. doi:10.1080/01441647.2014.903530.

[15] R. D. Das, S. Winter, Automated urban travel interpretation: A bottom-up approach for trajectory segmentation, Sensors (Switzerland) 16 (2016). doi:10.3390/s16111962.

[16] M. S. Iqbal, C. F. Choudhury, P. Wang, M. C. González, Development of origin-destination matrices using mobile phone call data,

Transportation Research Part C: Emerging Technologies 40 (2014). doi:10.1016/j.trc.2014.01.002.

[17] J. Chen, M. Bierlaire, Probabilistic multimodal map matching with rich smartphone data, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 19 (2015) 134–148. doi:10.1080/15472450.2013.764796.

[18] G. R. Jagadeesh, T. Srikanthan, Online Map-Matching of Noisy and Sparse Location Data with Hidden Markov and Route Choice Models, IEEE Transactions on Intelligent Transportation Systems (2017). doi:10.1109/TITS.2017.2647967.

[19] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, Y. Huang, Map-matching for low-sampling-rate GPS trajectories, Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09 (2009) 352. URL: http://portal.acm.org/citation.cfm?doid=1653771.1653820. doi:10.1145/1653771.1653820.

[20] J. Huang, S. Qiao, H. Yu, J. Qie, C. Liu, Parallel map matching on massive vehicle GPS data using MapReduce, Proceedings - 2013 IEEE International Conference on High Performance Computing and Communications, HPCC 2013 and 2013 IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2013 (2014) 1498–1503. doi:10.1109/HPCC.and.EUC.2013.211.

[21] N. Garg, Mining Bus Stops from Raw GPS Data of Bus Trajectories, in: 10th International Conference on Communication Systems & Networks (COMSNETS), IEEE, Bengaluru, India, 2018, pp. 583–588. doi:10.1109/COMSNETS.2018.8328278.

[22] A. C. Prelipcean, G. Gidófalvi, Y. O. Susilo, MEILI: A travel diary collection, annotation and automation system, Computers, Environment and Urban Systems 70 (2018). doi:10.1016/j.compenvurbsys.2018.01.011.

[23] P. Anderson, M. Hepworth, B. Kelly, R. Metcalfe, What is Web 2 . 0 ? Ideas , technologies and implications for education by, Technology 60 (2007) 64. URL: http://www.jisc.ac.uk/media/documents/techwatch/tsw0701b.pdf.

[24] P. Nitsche, P. Widhalm, S. Breuss, N. Brändle, P. Maurer, Supporting large-scale travel surveys with smartphones - A practical approach, Transportation Research Part C: Emerging Technologies 43 (2014) 212–221. URL: http://dx.doi.org/10.1016/j.trc.2013.11.005. doi:10.1016/j.trc.2013.11.005.

[25] P. R. Stopher, S. P. Greaves, Household travel surveys: Where are we going?, Transportation Research Part A: Policy and Practice 41 (2007) 367–381. doi:10.1016/j.tra.2006.09.005.

[26] F. Zhao, A. Ghorpade, F. C. Pereira, C. Zegras, M. Ben-Akiva, Stop detection in smartphone-based travel surveys, Transportation Research Procedia 11 (2015) 218–226. URL: http://dx.doi.org/10.1016/j.trpro.2015.12.019. doi:10.1016/j.trpro.2015.12.019.

[27] A. Ek, C. Alexandrou, C. Delisle Nyström, A. Direito, U. Eriksson, U. Hammar, P. Henriksson, R. Maddison, Y. Trolle Lagerros, M. Löf, The Smart City Active Mobile Phone Intervention (SCAMPI) study to promote physical activity through active transportation in healthy adults: A study protocol for a randomised controlled trial, BMC Public Health (2018). doi:10.1186/s12889-018-5658-4.

[28] C. Calastri, R. Crastes Dit Sourd, S. Hess, We want it all: experiences from a survey seeking to capture social network structures, lifetime events and short-term travel and activity planning, Transportation (2018). doi:10.1007/s11116-018-9858-7.

[29] Z. Patterson, K. Fitzsimmons, S. Jackson, T. Mukai, Itinerum: The open smartphone travel survey platform, SoftwareX 10 (2019) 100230. URL: https://doi.org/10.1016/j.softx.2019.04.002. doi:10.1016/j.softx.2019.04.002.

[30] Z. Patterson, K. Fitzsimmons, Datamobile: Smartphone travel survey experiment., Transportation Research Record 2594 (2016) 35–53.

[31] S. Greaves, A. Ellison, R. Ellison, D. Rance, C. Standen, C. Rissel, M. Crane, A web-based diary and companion smartphone app for travel/activity surveys, in: Transportation Research Procedia, 2015. doi:10.1016/j.trpro.2015.12.026.

[32] A. Yazdizadeh, Z. Patterson, B. Farooq, Ensemble Convolutional Neural Networks for Mode Inference in Smartphone Travel Survey, IEEE Transactions on Intelligent Transportation Systems (2019). doi:10.1109/tits.2019.2918923. arXiv:1904.08933.

[33] P. R. Stopher, L. Shen, W. Liu, A. Ahmed, The challenge of obtaining ground truth for gps processing, Transportation Research Procedia 11 (2015) 206–217. URL: https://www.sciencedirect.com/science/article/pii/S2352146515003099. doi:https://doi.org/10.1016/j.trpro.2015.12.018, transport Survey Methods: Embracing Behavioural and Technological Changes Selected contributions from the 10th International Conference on Transport Survey Methods 16-21 November 2014, Leura, Australia.

[34] B. Assemi, H. Jafarzadeh, M. Mesbah, M. Hickman, Participants' perceptions of smartphone travel surveys, Transportation Research Part F: Traffic Psychology and Behaviour 54 (2018) 338–348. URL: https://doi.org/10.1016/j.trf.2018.02.005. doi:10.1016/j.trf.2018.02.005.

[35] Apple, Preventing unexpected shutdowns, 2019. URL: https://support.apple.com/en-us/HT208387, retrieved from web 01/01/2020.

[36] Y. A. De Montjoye, C. A. Hidalgo, M. Verleysen, V. D. Blondel, Unique in the Crowd: The privacy bounds of human mobility, Scientific Reports 3 (2013) 1–5. doi:10.1038/srep01376.

[37] D. E. Seidl, P. Jankowski, M. H. Tsou, Privacy and spatial pattern preservation in masked GPS trajectory data, International Journal of Geographical Information Science 30 (2016) 785–800. URL: http://dx.doi.org/10.1080/13658816.2015.1101767. doi:10.1080/13658816.2015.1101767.

[38] V. Primault, A. Boutet, S. B. Mokhtar, L. Brunie, The long road to computational location privacy: A survey, Ieee Communications Surveys and Tutorials 21 (2019) 8482357, 2772–2793. doi:10.1109/COMST.2018.2873950.

[39] G. P. Perrucci, F. H. P. Fitzek, J. Widmer, Survey on energy consumption entities on the smartphone platform, in: 2011 IEEE

73rd Vehicular Technology Conference (VTC Spring), 2011, pp. 1–6. doi:10.1109/VETECS.2011.5956528.

[40] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, D. Roggen, Enabling reproducible research in sensor-based transportation mode recognition with the sussex-huawei dataset, IEEE Access (2019). doi:10.1109/ACCESS.2019.2890793.

[41] T. O. Oshin, S. Poslad, A. Ma, Improving the energy-efficiency of GPS based location sensing smartphone applications, Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012 (2012) 1698–1705. doi:10.1109/TrustCom.2012.184.

[42] Apple, Apple developers support resolution on network signal strength access, 2016. URL: https://forums.developer.apple.com/message/196395#196395, retrieved from web 01/01/2019.

[43] S. A. Hoseini-Tabatabaei, A. Gluhak, R. Tafazolli, A survey on smartphone-based systems for opportunistic user context recognition, ACM Computing Surveys 45 (2013) 1–51. URL: http://dl.acm.org/citation.cfm?doid=2480741.2480744. doi:10.1145/2480741.2480744.

[44] X. Li, X. Zhang, K. Chen, S. Feng, Measurement and analysis of energy consumption on android smartphones, in: 2014 4th IEEE International Conference on Information Science and Technology, 2014, pp. 242–245. doi:10.1109/ICIST.2014.6920375.

[45] A. Allström, I. Kristoffersson, Y. Susilo, Smartphone based based travel diary collection: experiences from a field trial in Stockholm, in: Transportation Research Procedia, volume 26, Elsevier B.V., Barcellona, 2017, pp. 32–38. URL: http://dx.doi.org/10.1016/j.trpro.2017.07.006. doi:10.1016/j.trpro.2017.07.006.

[46] C. Cottrill, F. Pereira, F. Zhao, I. Dias, H. Lim, M. Ben-Akiva, P. Zegras, Future Mobility Survey, Transportation Research Record: Journal of the Transportation Research Board 2354 (2013) 59–67. URL: http://trrjournalonline.trb.org/doi/10.3141/2354-07. doi:10.3141/2354-07.

[47] K. E. Jeon, J. She, P. Soonsawad, P. C. Ng, BLE Beacons for Internet of Things Applications: Survey, Challenges, and Opportunities, IEEE Internet of Things Journal 5 (2018) 811–828. doi:10.1109/JIOT.2017.2788449.

[48] P. Davidson, R. Piché, A survey of selected indoor positioning methods for smartphones, IEEE Communications Surveys Tutorials 19 (2017) 1347–1370.

[49] C. Li, P. C. Zegras, F. Zhao, Z. Qin, A. Shahid, M. Ben-Akiva, F. Pereira, J. Zhao, Enabling Bus Transit Service Quality Co-Monitoring Through Smartphone-Based Platform, Transportation Research Record: Journal of the Transportation Research Board 2649 (2017) 42–51. doi:10.3141/2649-05.

[50] L. Montini, N. Rieser-Schüssler, A. Horni, K. Axhausen, Trip Purpose Identification from GPS Tracks, Transportation Research Record: Journal of the Transportation Research Board (2014). doi:10.3141/2405-03. arXiv:arXiv:1011.1669v3.

[51] T. K. Rasmussen, J. B. Ingvardson, K. Halldórsdóttir, O. A. Nielsen, Improved methods to deduct trip legs and mode from travel surveys using wearable GPS devices: A case study from the Greater Copenhagen area, Computers, Environment and Urban Systems 54 (2015) 301–313. doi:10.1016/j.compenvurbsys.2015.04.001.

[52] N. E. E. Faouzi, H. Leung, A. Kurian, Data fusion in intelligent transportation systems: Progress and challenges - A survey, Information Fusion (2011). doi:10.1016/j.inffus.2010.06.001.

[53] S. Kanarachos, S. R. G. Christopoulos, A. Chroneos, Smartphones as an integrated platform for monitoring driver behaviour: The role of sensor fusion and connectivity, Transportation Research Part C: Emerging Technologies (2018) 0–1. URL: https://doi.org/10.1016/j.trc.2018.03.023. doi:10.1016/j.trc.2018.03.023.

[54] M. Kubicka, A. Cela, H. Mounier, S. I. Niculescu, Comparative Study and Application-Oriented Classification of Vehicular Map-Matching Methods, IEEE Intelligent Transportation Systems Magazine 10 (2018) 150–166. doi:10.1109/MITS.2018.2806630.

[55] M. Bierlaire, J. Chen, J. Newman, A probabilistic map matching method for smartphone GPS data, Transportation Research Part C: Emerging Technologies 26 (2013) 78–98. URL: http://dx.doi.org/10.1016/j.trc.2012.08.001. doi:10.1016/j.trc.2012.08.001.

[56] Y. J. Byon, 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: Technology, Planning, and Operations (2014). doi:10.1080/15472450.2013.824762.

[57] J. R. Blum, D. G. Greencorn, J. R. Cooperstock, Smartphone Sensor Reliability for Augmented Reality Applications, in: K. Zheng, M. Li, H. Jiang (Eds.), Mobile and Ubiquitous Systems: Computing, Networking, and Services, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 127–138.

[58] D. L. Silver, Q. Yang, L. Li, Lifelong machine learning systems: Beyond learning algorithms, in: 2013 AAAI spring symposium series, Citeseer, 2013.

[59] Y. Kim, F. C. Pereira, P. C. Zegras, M. Ben-akiva, Activity Recognition for a Smartphone and Web- Based Human Mobility Sensing System, IEEE Intelligent Systems 33 (2018) 5–23. doi:10.1109/MIS.2018.043741317.

[60] A. N. Koushik, M. Manoj, N. Nezamuddin, Machine learning applications in activity-travel behaviour research: a review, Transport Reviews 0 (2020) 1–24. URL: https://doi.org/10.1080/01441647.2019.1704307. doi:10.1080/01441647.2019.1704307. arXiv:https://doi.org/10.1080/01441647.2019.1704307.

[61] A. Abbruzzo, M. Ferrante, S. D. Cantis, A pre-processing and network analysis of gps tracking data, Spatial Economic Analysis 16 (2021) 217–240. URL: https://doi.org/10.1080/17421772.2020.1769170. doi:10.1080/17421772.2020.1769170. arXiv:https://doi.org/10.1080/17421772.2020.1769170.

[62] C. Velasco-Gallego, I. Lazakis, Real-time data-driven missing data imputation for short-term sensor data of marine systems. a comparative study, Ocean Engineering 218 (2020)

108261. URL: https://www.sciencedirect.com/science/article/pii/S0029801820311823. doi:https://doi.org/10.1016/j.oceaneng.2020.108261.

[63] D. Rolnick, A. Veit, S. Belongie, N. Shavit, Deep learning is robust to massive label noise (2018).

[64] D. F. Nettleton, A. Orriols-Puig, A. Fornells, A study of the effect of different types of noise on the precision of supervised learning techniques, Artificial Intelligence Review 33 (2010) 275–306. doi:10.1007/s10462-010-9156-z.

[65] E. Beigman, B. B. Klebanov, Learning with annotation noise, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1, ACL '09, Association for Computational Linguistics, USA, 2009, p. 280–287.

[66] N. Manwani, P. S. Sastry, Noise tolerance under risk minimization, IEEE Transactions on Cybernetics 43 (2013) 1146–1151. doi:10.1109/TSMCB.2012.2223460.

[67] C. Man Teng, A comparison of noise handling techniques (2015). doi:10.1.1.529.9973.

[68] R. Barandela, E. Gasca, Decontamination of training samples for supervised pattern recognition methods, in: F. J. Ferri, J. M. Iñesta, A. Amin, P. Pudil (Eds.), Advances in Pattern Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 621–630.

[69] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. De Macedo, B. Moelans, A. Vaisman, A model for enriching trajectories with semantic geographical information, in: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, 2007. doi:10.1145/1341012.1341041.

[70] Q. Zhu, M. Zhu, M. Li, M. Fu, Z. Huang, Q. Gan, Z. Zhou, Identifying transportation modes from raw GPS data, in: Communications in Computer and Information Science, 2016. doi:10.1007/978-981-10-2053-7\$_3$5.

[71] Y. Zheng, Trajectory data mining: An overview, ACM Trans. Intell. Syst. Technol. 6 (2015). URL: https://doi.org/10.1145/2743025. doi:10.1145/2743025.

[72] J. Van Dijk, Identifying activity-travel points from GPS-data with multiple moving windows, Computers, Environment and Urban Systems 70 (2018) 84–101. URL: https://doi.org/10.1016/j.compenvurbsys.2018.02.004. doi:10.1016/j.compenvurbsys.2018.02.004.

[73] N. Schuessler, K. W. Axhausen, Processing raw data from global positioning systems without additional information, Transportation Research Record 2105 (2009) 28–36. URL: https://doi.org/10.3141/2105-04. doi:10.3141/2105-04.

[74] R. Ehsani, S. Buchanon, M. Salyani, Gps accuracy for tree scouting and other horticultural uses, EDIS 2009 (2009). URL: https://journals.flvc.org/edis/article/view/117815.

[75] S. von Watzdorf, F. Michahelles, Accuracy of positioning data on smartphones, in: Proceedings of the 3rd International Workshop on Location and the Web, LocWeb '10, Association for Computing Machinery, New York, NY, USA, 2010. URL: https://doi.org/10.1145/1899662.1899664. doi:10.1145/1899662.1899664.

[76] M. Rosvall, D. Axelsson, C. T. Bergstrom, The map equation, European Physical Journal Special Topics 178 (2009) 13–23. doi:10.1140/epjst/e2010-01179-1. arXiv:0906.1405.

[77] U. Aslak, Infostop, a Python package for detecting stop locations in mobility data, 2019. URL: https://github.com/ulfaslak/infostop, retrieved from web 26/11/2019.

[78] A. Tietbohl, V. Bogorny, B. Kuijpers, L. O. Alvares, A clustering-based approach for discovering interesting places in trajectories, in: Proceedings of the ACM Symposium on Applied Computing, 2008. doi:10.1145/1363686.1363886.

[79] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from GPS trajectories, in: Proceedings of the

18th international conference on World wide web - WWW '09, 2009. doi:10.1145/1526709.1526816.

[80] R. Guidotti, R. Trasarti, M. Nanni, TOSCA: TwO-Steps Clustering Algorithm for personal locations detection, in: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, 2015. doi:10.1145/2820783.2820818.

[81] L. Xiang, M. Gao, T. Wu, Extracting stops from noisy trajectories: A sequence oriented clustering approach, ISPRS International Journal of Geo-Information (2016). doi:10.3390/ijgi5030029.

[82] D. Wang, J. Zhang, W. Cao, J. Li, Y. Zheng, When Will You Arrive ? Estimating Travel Time Based on Deep Neural Networks, Ijcai (2018).

[83] B. Thierry, B. Chaix, Y. Kestens, Detecting activity locations from raw GPS data: A novel kernel-based algorithm, International Journal of Health Geographics (2013). doi:10.1186/1476-072X-12-14.

[84] R. Hariharan, K. Toyama, Project lachesis: Parsing and modeling location histories, in: M. J. Egenhofer, C. Freksa, H. J. Miller (Eds.), Geographic Information Science, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 106–124.

[85] P. Nurmi, J. Koolwaaij, Identifying meaningful locations, in: 2006 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous, 2006. doi:10.1109/MOBIQ.2006.340429.

[86] G. Xiao, Q. Cheng, C. Zhang, Detecting travel modes from smartphone-based travel surveys with continuous hidden Markov models, International Journal of Distributed Sensor Networks (2019). doi:10.1177/1550147719844156.

[87] L. Liao, D. Fox, H. Kautz, Extracting places and activities from GPS traces using hierarchical conditional random fields, International Journal of Robotics Research (2007). doi:10.1177/0278364907073775.

[88] R. Zhou, M. Li, H. Wang, X. Song, W. Xie, Z. Lu, An Enhanced Transportation Mode Detection Method Based on GPS Data, in: Communications in Computer and Information Science, volume 727, 2017, pp. 605–620. doi:10.1007/978-981-10-6385-5_51.

[89] S. Dabiri, K. Heaslip, Inferring transportation modes from GPS trajectories using a convolutional neural network, Transportation Research Part C: Emerging Technologies 86 (2018) 360–371. URL: https://doi.org/10.1016/j.trc.2017.11.021. doi:10.1016/j.trc.2017.11.021.

[90] X. Jiang, E. N. de Souza, A. Pesaranghader, B. Hu, D. L. Silver, S. Matwin, TrajectoryNet: An Embedded GPS Trajectory Representation for Point-based Classification Using Recurrent Neural Networks, 2017. URL: http://arxiv.org/abs/1705.02636. arXiv:1705.02636, source code published on Github @ https://github.com/wuhaotju/TrajectoryNet. Retrieved from web 01/11/2019.

[91] A. C. Prelipcean, G. Gidofalvi, Y. O. Susilo, Measures of transport mode segmentation of trajectories, International Journal of Geographical Information Science 30 (2016) 1763–1784. URL: http://dx.doi.org/10.1080/13658816.2015.1137297. doi:10.1080/13658816.2015.1137297. arXiv:arXiv:1505.06786v1.

[92] J. F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM 26 (1983) 832–843. URL: http://portal.acm.org/citation.cfm?doid=182.358434. doi:10.1145/182.358434.

[93] G. Vuk, J. L. Bowman, A. Daly, S. Hess, Impact of family in-home quality time on person travel demand, Transportation 43 (2016) 705–724. URL: https://doi.org/10.1007/s11116-015-9613-2. doi:10.1007/s11116-015-9613-2.

[94] Y. Zheng, H. Fu, Geolife GPS trajectory dataset - User Guide, Technical Report November 31, 2011. URL: http://research.microsoft.com/apps/pubs/?id=152176{%}5Cnhttp://research.microsoft.com/apps/pubs/default.aspx?id=152176, online; accessed 19-July-2008.

[95] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: An update, SIGKDD Explor. Newsl. 11 (2009). URL: https://doi-org.proxy.findit.dtu.dk/10.1145/1656274.1656278. doi:10.1145/1656274.1656278.

[96] W. Ectors, S. Reumers, W. D. Lee, K. Choi, B. Kochan, D. Janssens, T. Bellemans, G. Wets, Developing an optimised activity type annotation method based on classification accuracy and entropy indices, Transportmetrica A: Transport Science 13 (2017) 742–766. URL: https://doi.org/10.1080/23249935.2017.1331275. doi:10.1080/23249935.2017.1331275. arXiv:https://doi.org/10.1080/23249935.2017.1331275.

[97] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, S. M. Wild, DeepHyper: Asynchronous Hyperparameter Search for Deep Neural Networks, Proceedings - 25th IEEE International Conference on High Performance Computing, HiPC 2018 (2019) 42–51. doi:10.1109/HiPC.2018.00014.

[98] L. Xiao, Y. Li, G. Han, H. Dai, H. V. Poor, A Secure Mobile Crowdsensing Game with Deep Reinforcement Learning, IEEE Transactions on Information Forensics and Security 13 (2018) 35–47. doi:10.1109/TIFS.2017.2737968.

[99] G. Sicotte, C. Morency, B. Farooq, Comparison between trip and trip chain models: Evidence from montreal commuter train corridor, 2017. doi:10.13140/RG.2.2.21494.80963.

[100] M. Cornacchia, K. Ozcan, Y. Zheng, S. Velipasalar, A survey on activity detection and classification using wearable sensors, Ieee Sensors Journal 17 (2017) 7742959. doi:10.1109/JSEN.2016.2628346.

[101] Apple, Car data integration on smartphones, 2021. URL: https://developer.apple.com/design/human-interface-guidelines/carplay/interaction/car-data/, retrieved from web 17/03/2021.

[102] L. Wang, L. Jiao, J. Li, J. Gedeon, M. Mühlhäuser, Moera: Mobility-agnostic online resource allocation for edge computing, IEEE Transactions on Mobile Computing 18 (2019) 1843–1856. doi:10.1109/TMC.2018.2867520.

[103] R. Bellman, Dynamic programming, Princeton University Press,, 1957.

[104] K. Shankari, J. Fuerst, M. F. Argerich, E. Avramidis, J. Zhang, Mobilitynet: Towards a public dataset for multi-modal mobility research (2020).

[105] M. Kubicka, A. Cela, P. Moulin, H. Mounier, S. I. Niculescu, Dataset for testing and training map-matching methods, 2016. URL: https://doi.org/10.5281/zenodo.57731. doi:10.5281/zenodo.57731.

[106] C. Carpineti, V. Lomonaco, L. Bedogni, M. D. Felice, L. Bononi, Custom dual transportation mode detection by smartphone devices exploiting sensor diversity, Proc. of the 14th Workshop on Context and Activity Modeling and Recognition (IEEE COMOREA 2018) (2018). doi:https://doi.org/10.1109/PERCOMW.2018.8480119.

[107] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. del R. Millán, D. Roggen, The opportunity challenge: A benchmark database for on-body sensor-based activity recognition, Pattern Recognition Letters 34 (2013) 2033–2042. URL: https://www.sciencedirect.com/science/article/pii/S0167865512004205. doi:https://doi.org/10.1016/j.patrec.2012.12.014, smart Approaches for Human Action Recognition.

[108] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T. M. T. Do, O. Dousse, J. Eberle, M. Miettinen, From big smartphone data to worldwide research: The mobile data challenge, Pervasive and Mobile Computing 9 (2013) 752–771. doi:10.1016/j.pmcj.2013.07.014.

[109] R. P. Baker, N. M. Bradburn, R. A. Johnson, Computer-assisted personal interviewing: an experimental evaluation of data quality and cost, Journal of Official Statistics 11 (1995) 413–431.

[110] L. Nicholls II, R. M. Groves, The status of computer-assisted telephone interviewing: Part i-introduction and impact on cost and timeliness of survey data, Journal of official statistics 2 (1986) 93.

[111] J. Zmud, M. Lee-Gosselin, J. A. Carrasco, M. A. Munizaga, Transport survey methods: Best practice for decision making, Emerald Group Publishing, 2013.

[112] F. Zhao, F. C. Pereira, R. Ball, Y. Kim, Y. Han, C. Zegras, M. Ben-Akiva, Exploratory analysis of a smartphone-based travel survey in singapore, Transportation Research Record 2494 (2015) 45–56. URL: https://doi.org/10.3141/2494-06. doi:10.3141/2494-06. arXiv:https://doi.org/10.3141/2494-06.

[113] L. Christensen, The role of web interviews as part of a national travel survey, Transport Survey Methods: Best Practice for Decision Making (2013) 115–153.

[114] W. Bohte, K. Maat, Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: A large-scale application in the Netherlands, Transportation Research Part C: Emerging Technologies 17 (2009) 285–297. URL: http://dx.doi.org/10.1016/j.trc.2008.11.004. doi:10.1016/j.trc.2008.11.004.

[115] B. D. Martin, V. Addona, J. Wolfson, G. Adomavicius, Y. Fan, Methods for real-time prediction of the mode of travel using smartphone-based GPS and accelerometer data, Sensors (Switzerland) 17 (2017). doi:10.3390/s17092058.

[116] A. Jahangiri, H. A. Rakha, Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data, IEEE Transactions on Intelligent Transportation Systems 16 (2015) 2406–2417. doi:10.1109/TITS.2015.2405759.

[117] X. Zhou, W. Yu, W. C. Sullivan, Making pervasive sensing possible: Effective travel mode sensing based on smartphones, Computers, Environment and Urban Systems 58 (2016) 52–59. URL: http://dx.doi.org/10.1016/j.compenvurbsys.2016.03.001. doi:10.1016/j.compenvurbsys.2016.03.001.

[118] X. Zhu, J. Li, Z. Liu, S. Wang, F. Yang, Learning transportation annotated mobility profiles from GPS data for context-aware mobile services, Proceedings - 2016 IEEE International Conference on Services Computing, SCC 2016 (2016) 475–482. doi:10.1109/SCC.2016.68.

[119] G. Xiao, Z. Juan, C. Zhang, Travel mode detection based on GPS track data and Bayesian networks, Computers, Environment and Urban Systems 54 (2015) 14–22. doi:10.1016/j.compenvurbsys.2015.05.005.

[120] A. Yazdizadeh, Z. Patterson, B. Farooq, An automated approach from GPS traces to complete trip information, International Journal of Transportation Science and Technology (2019). doi:10.1016/j.ijtst.2018.08.003.

[121] S. Dabiri, C.-T. Lu, K. Heaslip, C. K. Reddy, Semi-Supervised Deep Learning Approach for Transportation Mode Identification Using GPS Trajectory Data, IEEE Transactions on Knowledge and Data Engineering (2019). doi:10.1109/tkde.2019.2896985.

[122] B. Assemi, H. Safi, M. Mesbah, L. Ferreira, Developing and Validating a Statistical Model for Travel Mode Identification on Smartphones, IEEE Transactions on Intelligent Transportation Systems 17 (2016) 1920–1931. doi:10.1109/TITS.2016.2516252.

[123] H. Mäenpää, A. Lobov, J. L. Martinez Lastra, Travel mode estimation for multi-modal journey planner, Transportation Research Part C: Emerging Technologies 82 (2017) 273–289. doi:10.1016/j.trc.2017.06.021.

[124] A. Yazdizadeh, Z. Patterson, B. Farooq, Semi-supervised GANs to Infer Travel Modes in GPS Trajectories, 2019. URL: http://arxiv.org/abs/1902.10768. arXiv:1902.10768.

[125] T. Feng, H. J. Timmermans, Detecting activity type from gps traces using spatial and temporal information, European Journal of Transport and Infrastructure Research 15 (2015).

[126] F. Torre, D. Pitchford, P. Brown, L. Terveen, Matching GPS traces to (possibly) incomplete map data, Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12 (2012) 546. URL: http://dl.acm.org/citation.cfm?doid=2424321.2424411. doi:10.1145/2424321.2424411.

[127] M. A. Quddus, R. B. Noland, W. Y. Ochieng, A high accuracy fuzzy logic based map matching algorithm for road transport, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 10 (2006) 103–115. doi:10.1080/15472450600793560.

[128] L. Li, M. Quddus, L. Zhao, High accuracy tightly-coupled integrity monitoring algorithm for map-matching, Transportation Research Part

C: Emerging Technologies 36 (2013) 13–26. URL: http://dx.doi.org/10.1016/j.trc.2013.07.009. doi:10.1016/j.trc.2013.07.009.

[129] H. Wei, Y. Wang, G. Forman, Y. Zhu, Map matching: Comparison of approaches using sparse and noisy data, in: Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL'13, Association for Computing Machinery, New York, NY, USA, 2013, p. 444–447. URL: https://doi.org/10.1145/2525314.2525456. doi:10.1145/2525314.2525456.

[130] H. Wu, Z. Chen, W. Sun, B. Zheng, W. Wang, Modeling trajectories with recurrent neural networks, in: IJCAI International Joint Conference on Artificial Intelligence, 2017, pp. 3083–3090. doi:10.24963/ijcai.2017/430.

[131] T. Hunter, P. Abbeel, A. Bayen, The path inference filter: Model-based low-latency map matching of probe vehicle data, IEEE Transactions on Intelligent Transportation Systems 15 (2014) 507–529. doi:10.1109/TITS.2013.2282352. arXiv:1109.1966.

[132] H. Li, G. Wu, Map Matching for Taxi GPS Data with Extreme Learning Machine, volume 8933, Springer, 2014.

[133] P. Newson, J. Krumm, Hidden Markov map matching through noise and sparseness, Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09 (2009) 336–343. URL: http://portal.acm.org/citation.cfm?doid=1653771.1653818. doi:10.1145/1653771.1653818.

3 Paper B: Stop detection for smartphone-based travel surveys using geo-spatial context and artificial neural networks

The following pages contain the article:

V. Servizi, N. C. Petersen, F. C. Pereira, and O. A. Nielsen (2020). "Stop detection for smartphone-based travel surveys using geo-spatial context and artificial neural networks". In: *Transportation Research Part C: Emerging Technologies* 121, p. 102834. DOI: 10.1016/j.trc.2020.102834. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X20307385.

Please cite accordingly.

The work was part of poster presentations at the *"8th Symposium of the European Association for Research in Transportation (hEART 2019)"*, Budapest September, 2019.

Stop Detection for Smartphone-based travel surveys using Geo-spatial context and Artificial Neural Networks

Valentino Servizi*, Niklas C. Petersen, Francisco C. Pereira, Otto A. Nielsen

Department of Management Engineering
Technical University of Denmark (DTU)
Kgs. Lyngby Denmark

Abstract

The problem of stop detection is at the base of many current and upcoming smartphone-based travel survey technologies and directly impacts the quality of many downstream operations. The inference of departure/arrival time, mode, and purpose of a trip, for example, rely on the stop/motion patterns represented by smartphone sensors data. As users handle smartphones for various purposes and their preferences determine different device positions while traveling, accelerometer, and gyroscope, for instance, often present ambiguities that prevent accurate stop detection.

To mitigate the impact of these ambiguities, we combine spatial time-series, i.e. GPS, with spatial context information retrieved from Open Street Map, which we represent as multi-dimension tensors. This project explores simple representations, such as dummy variables, and novel multidimensional representations, which are bench-marked through the classification performance of specialized artificial neural network (ANN), as well as other machine learning (ML) baselines. Our main contribution stems from this novel multidimensional representation of time-series fusion with spatial context, combined with the corresponding specialized ANN classifier. The results show a stop detection score improvement on the baselines between 3% and 6.5%.

Keywords: GPS+GIS fusion, stop detection, smartphone based travel surveys, ANN, CNN, RNN, point based classification, GPS, trajectories

*Corresponding author. Email: valse@dtu.dk

1. Introduction

Smartphone-based travel surveys' (SBTS) study of users behavior in transport networks, relies heavily on GPS trajectories.

As people *move* for a purpose and *stop* to fulfill that purpose, we define the subset of a GPS trajectory where the user travels on any transportation mode between origin and destination, as *motion*; everything else, we define as *stop*.

Stemming from SBTS, literature on mode detection shows how motion branches out; literature on purpose imputation, how stop branches out. Both deal with trip segmentation, and contribute to the automatic generation of Travel Diaries (TDs) that are presented to users for ground truth collection, within a continuous *validation loop* [1]. In this cycle, ground truth supposedly improves machine learning algorithms for TDs generation, and vice-versa.

However, in terms of GPS point density, short-duration stops, such as a pick-up, or a drop-off, are substantially different than stops with a longer duration, such as home or work stay. The same applies to motion; for example, when one moves by car versus walking.

In many cases the two categories result entangled on both time and space. Let us consider two representative examples:

- Alighting from a bus is often seen as an *instant stop* since one changes transportation mode from bus to walk; nevertheless, during this transition the user is never stationary.
- During a bus trip, discontinuities at traffic lights have rarely short duration, as at each bus stop between origin and destination. However, there is no transition between transportation modes.

In addition, because of a standard deviation above 40m [2, 3], GPS error itself can confuse inference, depending on building surroundings, atmospheric conditions, or relative satellite positioning, worsening the challenge of points' or segments' classification.

As [4] shows, these aspects could explain the severe ambiguities in GPS trajectories collected by origin-destination surveys, especially in short transfers.

Besides, the two classes of *stop* and *motion* suffer from a high imbalance since one moves only a small fraction of the day. In Denmark, for example, the average travel time measured in minutes/person/day is 57min [5].

Thus, the stop class might overwhelm any ML classifier, resulting in poor classification performance.

A wide range of classifiers is available in the SBTS field [6]. We identify two groups, point- and segment-based classifiers [7]. In the first case, these methods classify each GPS position; in the second, a segment composed of multiple GPS points. Among the most common trajectory segmentation methods for stop classification, rule-based methods seem very competitive [8]; we also find various practical clustering approaches [9].

Literature has shown that existing methods suffer the entanglement of these two classes, as both thresholds definition and features engineering are not effective in catching both short stops and long discontinuities during motion. Inaccurate classification of *stops*, leads to trip over- or under-segmentation; thus, to automatic generation of inexact TDs. In case of over-segmentation, TDs will present at least one true trip-leg split in two or more classified trip-legs, while in case of under-segmentation, TDs will present at least two true trip-legs merged into one classified trip-leg.

The under-segmentation problem occurs when all the points of a stops are classified as motion; the over-segmentation, when a number of consecutive motion points are classified as stops, or viceversa. Under-segmentation bias, e.g., in correspondence of instant-stops, is critical for users' TDs validation. In the assumption that one remembers such an instant-stop and is very committed to the survey, he or she should manually add any missing stop and the activity preformed within each of the corresponding space-time ranges.

Over-segmentation errors are as tedious to correct as the under-segmentation ones [3]. Both these errors might lead to unacceptable ground truth [7]. In light of the above considerations, how can we improve the discrimination between stop and motion by processing GPS trajectories?

To succeed, many machine learning (ML) methods exploit multiple sensors, and often geographical information systems (GIS). Among the best-performers, we find support vector machines (SVM), fuzzy logic (FL), random forests (RF), and probabilistic models, e.g., hidden markov models (HMM) [1].

In contrast, emerging methods based on ANN as in [10], [11], and [12], show classification potential due to their flexibility in learning multiple thresholds from multi-dimension tensor representations, e.g., images. However, ANN methods seldom combine GPS with GIS data, possibly with dummy variables, never through richer tensor representations, never for mode or stop detection.

In order to deal with the binary classification of GPS trajectories, including challenging short stops and long discontinuities detection in realistic datasets, we incorporate spatial context information, such as public transport network, points of interest (POI), and land use. Thus, fusing spatial features retrieved from GIS with GPS, we help better capture heterogeneous temporal and spatial thresholds through various ANN configurations.

The paper analyzes multiple ANN classifiers, specialized for point-based classification of trajectories fused on GIS data as multi-dimension tensor representations. Intuitively, these representations describe spatial-context as images; whereas dummy variables, as single-pixel images. We compare ANN against two ML classifiers. First, Infostop [9], which is an unsupervised effective hybrid rule- and clustering-based method; second, a random forest, which the literature describes as one of the most effective supervised classifiers. Every classifier, except Infostop, is tested with two data representations: GPS features only, and GPS features augmented with dummy variables extracted from GIS. ANN models, as part of our contribution, allow tests with a third representation of data, which is GPS features augmented with multi-dimension-dummy variables, as tensor, extracted from GIS.

We work with a high-resolution realistic dataset which includes shared mobility trips, where users are committed in high quality ground truth collection. In Sec. 2, we present the literature review on GPS stop detection, and we position of our contribution within SBTS and ANN related work. In Sec. 3, we describe in detail our contribution in terms of methodology for fusion of GPS with GIS data, and the ANN classifiers. In Sec. 4 we present and discuss input data, results, validaiton process and benchmark. We conclude in Sec. 5, including future directions.

2. Related work

In the following sections we describe literature on stop detection method, and we pinpoint what pertains ANN.

2.1. Review of existing stop detection methods

A GPS segment is considered a stop candidate if it lays within a topologically closed polygon for a certain time [13]. Since the introduction of GPS travel surveys, the identification of trips by stop detection has presented multiple challenges. For example, the GPS device's cold-start requires over 1 minute to acquire the device position, during which this device generates

erroneous data, easy to confuse with short-stops [14]. The literature, however, presents simple and effective rule-based methods and heuristics, able to classify GPS positions without any need for labels, which in contrast, are necessary for more advanced supervised machine learning methods. For instance, we could be looking at a stop [14] each time points have (i) null speed, (ii) null or unchanged bearing, (iii) following positions that present a change below 15 meters in either latitude or longitude, (iv) and all these conditions persist for 120 seconds. In smartphones, the cold-start problem has been mitigated by introducing, e.g., the Assisted-GPS (GPS), which retrieves the satellites' position from Internet, and estimates the device position triangulating the GSM signal strength received from the antennas in proximity, each time GPS satellites are not in sight [1]. This solution, however, comes at the cost of a larger GPS error (see Sec. 1). In this scenario, a broadly applied heuristic considers whether consecutive GPS positions, within a 1-minute range, persist or not within a spatial range of 50 meters [15]. The literature shows many developments in this direction, employing clustering techniques [16, 17, 18, 19], which can learn unsupervised and find stops within GPS trajectories. In multiple-step approaches, personal- [18], and geographical-context [17] can augment trajectories' information and improve stops candidates' classification. Density-based spatial clustering of applications with noise (DBSCAN) is at the base of most frameworks; some of these frameworks can even find stop candidates directly on raster image representations [20]. Many other effective probabilistic unsupervised methods are available, as for example kernel-based [21, 22], generative [23, 24], and discriminative [25], such as kernel-density algorithms, hidden markov models, and conditional random fields.

Some of these methods can be implemented to learn supervised by labels, or combined in multi-step approaches. Among the resulting hybrid solutions, we mention the integration of DBSCAN with a support vector machines [26], where the latter is a supervised-method. Within the realm of SBTS, Zhao et al. [27] describes stop detection in Future Mobility Survey (FMS), which is a popular sensing platform for prompted recall surveys. The method uses a basic space/time-range set of rules for finding stop-candidates. Then, it retrieves frequent-place signatures from users' personal information, it merges continuous stops with still-mode, and it removes stops in excess after mode detection. Still-mode is an entity closely correlated with stops. To detect this class as one of a larger set of modes, this classifier is trained by labels. Using SBTS data and labels, thus a supervised ML method, a random forest can

be very effective in still-mode detection and other transportation modes [28].

Methods that use GPS trajectories can rely on derived features, such as: distance, speed, acceleration, bearing change, and time intervals. These features come in multiple flavors and are strictly related to the motion/still states of the device [1, 29, 30]. Time also correlates to human activities, as work and leisure; thus, time of day, time of the week, time of the month, and time of year bring valuable information [1, 29, 30]. Personal- and spatial-context data, which are very informative on where and why people travel, can improve methods performance. In the former category, we have features such as home and workplace addresses, socio-demographics, and trips history; in the latter category, e.g., land-use, nearby bus stops, routes, road, and rail network [1, 29, 30].

Depending on the available information within our application scenario, although supervised methods are proven very successful, in practice, collecting reliable labels from large users' populations, is hardly manageable with SBTS. On the contrary, unsupervised methods are often applied but do not provide the required level of accuracy. Therefore, they are combined with other methods. From this perspective, ANN have massive potential, as they can learn both supervised [10], semi-supervised [31], and unsupervised [32]. Furthermore, some of the ANN configurations that are proven very effective in multidimensional data representations, such as convolutional neural networks (CNN), could benefit from a richer representation of the spatial context, which so far is limited to dummy variables [11]. For further detail on work related to ANN, see Sec. 2.2.

2.2. ANN trip classifiers and corresponding data representations

We describe further relevant literature on ANN classifiers, by positioning our work, within the space that each of the following subsections identify.

2.2.1. ANN configurations

Artificial Neural Networks are a very successful parametric nonlinear modeling approach, for pattern recognition and classification, that maps a tensor $\mathbf{X}$ of input variables to a tensor $\mathbf{Y}$ of target variables. The subset of models we use in this work, includes Feed-Forward Networks (FFN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). These network models may specialize in different data structures.

FFN incorporate multiple layers of logistic regression with continuous and discontinuous nonlinearities. CNN are invariant to certain transformations of

the input, like translations, scaling, small rotations, and elastic deformations. RNN have the property of taking into account inputs processed in preceding chunks of a sequence, while producing the output from the last input. CNN are widely applied to images; RNN, to time series.

An ANN can result from a combination of sub-models as FFN, CNN, and RNN. Each model can be determined choosing a number of hyperparameters (HP). The resulting model's optimal parameters are specified based on a training dataset, by minimizing the loss function between the output of the model and the values of the target variable corresponding to the same input. In our case, FFN enable compounding the outputs of the preceding RNN and CNN models.

However, the likelihood function at the basis of ANN training is not a convex function of the model parameters. Consequently, training these models requires the allocation of substantial computational resources. Specifying the network parameters within a maximum likelihood framework involves the solution of a nonlinear optimization problem leveraging on backpropagation, which is a gradient-based algorithm. Therefore, to find a sufficiently good minimum, we need a grid-search and run such an algorithm multiple times, each time using a different combination of hyperparameters. The resulting performance, evaluated after training the model on an independent validation dataset, determines which hyper-parameters represent the best combination to solve the problem [33, Ch. 5], in this case, the classification of stop and move points.

2.3. Classification score and setup

F1-score can measure the performance of a classifier. It is the harmonic mean of precision and recall $\mathrm{F1} = 2\frac{P \cdot R}{P+R}$, where precision $P = \frac{T_p}{T_p+F_p}$, recall $R = \frac{T_p}{T_p+F_n}$. T_p stands for *True Positives*, e.g. stops classified as stops when stop is the target class, and motion classified as motion when motion is the target class. F_p stands for *False Positives*, e.g. motion points classified as stop points when stop is the target class, vice-versa when motion is the target class. F_n stands for *False Negatives*, e.g. stop points classified as motion points when stop is the target class, vice-versa when motion is the target class. In [10], F1 scores are the weighted average calculated on 5 classes, and on 5-fold cross validation. In [12], the authors pick a random sample of the users to compose training, validation, and test set, then F1 score is computed on 4 classes, and on the test set only. We call this 1-fold cross validation method *leave-one-out*.

In this work, we split the dataset in three partitions, and we leave one out. We find optimal hyperparameters with the remaining two partitions, which we call training (TR) and validation (VA). To estimate the overall error distribution, we fix the optimal hyperparameters, and we proceed in two steps. First, we estimate performance on the unseen partition, which we call test (TE). Second, we perform a k-fold validation on the whole dataset.

2.4. Classification of GPS representations

We refer to two papers on mode detection. In [12] a Recurrent Neural Network (RNN) classifies points over four modes, which are: walk, bus, bike, and car. This network is composed by a two layers bidirectional gated recurrent unit (GRU) with maxout activation function, which process discretized speed features after an embedding layer. Dabiri and Heaslip [10] present a CNN that classifies GPS segments over five modes: walk, bus, bike, car, and rail. Three convolutional layers using Rectified Linear Unit (ReLU) activation function compose the network, which process a four channel tensor representing: bearing rate, speed, acceleration and jerk. None of these two works implement data fusion with GIS. Referring to [10], a CNN approach seems to record the best performance over multiple baselines computed on the same settings, but with different machine learning methods, which are: (i) K-Nearest neighbor (KNN), with a range of neighbors between 3 and 40, finding 5 as optimal value; (ii) SVM with regularization in the range 0.5 and 20, finding 4 as optimal value; (iii) Decision tree (DT), with maximum depth of tree between 1 and 40, finding 10 as optimal value; (iv) Random forests (RF), with number of trees between 5 and 100, finding 85 as optimal value; (v) FFN, with number of hidden layers between 1 and 10, finding 1 as optimal value. Dabri et al. extends the work on ANN classifier in [31], so that the training can be semi-supervised.

2.5. Classification of GPS+GIS fusion representations

For purpose imputation, which is the classification of the activities performed at the stops (see Sec. 1), we found an example in [11]. The paper presents an FFN classifying a feature vector that includes land-use type (LT) and points of interest, coded as dummy variables. For ANN classifiers specialized in mode detection and stop detection, we did not find examples of GPS+GIS fusion. In general, we found no examples of GPS+GIS fusion with representations beyond the dummy variables' space. As already mentioned in the Sec. 1, smartphones are equipped with multiple sensors. Although

data recorded from these sensors can be fused on the time dimension [29] with GPS, we do not perform any fusion with other sensors at this time.

2.6. Classification approach

We found two main approaches, which are point-based and segment-based classification. As the name suggests, the entities that are processed and classified in the first case, are the features observed at each time step, while in the second case are sequences of observations [7] on the time dimension. To allow the comparison of competing classifiers, [7] proposed a penalization method to link the F1 score with the distances represented by the classification errors. By applying such a method, the authors show that point-based classifiers are superior than segment-based classifiers. Therefore, we choose to focus on a point-based classifier.

2.7. Comparability between different methods

There is consensus in the field about the lack of standardization for validating and comparing the performance of competing classifiers. [12] and [10] represent an evident example. Even though classifications are performed on the same dataset, they present different complexity as one deals with four classes, and the other with five; the validation setup is also different (see Sec. 2.3). Therefore, F1 scores comparison between these two tasks is meaningless. To mitigate the problem, in Sec. 2.6 we find a solution proposed by [7]; in [29], the authors propose both dataset and workflow for k-fold cross validation, which could provide a standardized baseline. Similarly to [29], to estimate the distribution of the classification errors, we fix the models' optimal HP, and we apply the same experiment setup to ANN methods and baselines. Thus, we compare F1-average distributions.

3. Methodology

In this section we describe our two main contributions: the fusion between GPS and GIS with a multi-dimension tensor, and the ANN configurations specialized in the classification of such a tensor.

3.1. Data requirements

Our approach rely on two datasets: *User GPS trajectories* and *Geospatial context data from Open Street Map* (OSM). While the latter is a global defined dataset, the GPS trajectories will in most use cases origin

from a geographically constrained area, e.g. country, or region where the travel survey is conducted. It is important that the two datasets cover the same range both with respect to space and time. For example, obviously, the subset of OSM used should cover the geographical area of the survey. To be meaningful, however, the OSM data should further describe a spatial context within the same *time frame* in which users generated their trajectories, as the geo-spatial context should not be altered.

We assume to have GPS trajectories for U different users, and for each user $u \in \{1, \ldots, U\}$ we denote the i^{th} GPS point, with $i \in \{1, \ldots, N_u\}$ having the following information:

- $t_{u,i}$: Time of the GPS point.
- $(lat_{u,i}, long_{u,i})$: Position components of the GPS point.
- $br_{u,i}$: Bearing between the GPS point i and $i-1$ in radians [10].
- $d_{u,i}$: Distance between GPS point i and $i-1$ in meters.

We denote the total number of GPS points $N = \sum_{u=1}^{U} N_u$. We further define the temporal context of each position using a *dummy variable* (1), as the discretization of $t_{u,i}$, taking into account that there is a significant difference between trip distributions during evening and night compared to the rest of the day [34]:

$$\mathbf{ToD}_{u,i} = \begin{cases} [1,0,0,0,0], & \text{if } t_{u,i} \in (00{:}00, 06{:}00] \\ [0,1,0,0,0], & \text{if } t_{u,i} \in (06{:}00, 10{:}00] \\ [0,0,1,0,0], & \text{if } t_{u,i} \in (10{:}00, 14{:}00] \\ [0,0,0,1,0], & \text{if } t_{u,i} \in (14{:}00, 18{:}00] \\ [0,0,0,0,1], & \text{if } t_{u,i} \in (18{:}00, 00{:}00] \end{cases} \tag{1}$$

The geo-spatial context, based on Open Streep Map [35], is a graph data model consisting of three basic data structures: nodes, ways, and relations. Each of these can represent physical features as shapes. For example, roads as segments, buildings and land use as polygons, intersections as points. Unique tags pinpoint each feature. We consider each specific value of a tag as distinct type of geo-spatial feature, weather they are attached to nodes, ways, or relations. We capture a selected subset of these feature types for use as the surrounding geo-spatial context of a given GPS trajectory point. The number of selected feature types is denoted C.

3.2. Data fusion process

To enrich GPS trajectories, we fuse each GPS point on the surrounding geo-spatial context, considering Open Street Map features within proximity of the GPS point. The result of the data fusion process is a multi-dimension tensor corresponding to each GPS position, $\mathbf{I}_{u,i} \in \mathbb{R}^{W \times H \times C}$. W and H represent the extension of the geo-spatial context in respectively horizontal and vertical direction from $(lat_{u,i}, long_{u,i})$ in some fixed unit (e.g. 10m steps), and C is the number of geo-spatial context features captured. The spacial center of the $\mathbf{I}_{u,i}$ tensor is always the corresponding GPS data point. If a specific geo-spatial feature, c, is present in proximity j, k from $(lat_{u,i}, long_{u,i})$ then $\mathbf{I}_{u,i,j,k,c} = 1$, otherwise $\mathbf{I}_{u,i,j,k,c} = 0$.

However, naïvely querying for each GPS data point, $(lat_{u,i}, long_{u,i})$ to contruct $\mathbf{I}_{u,i}$, is not computationally feasible. Hence, we pre-compute the geo-spatial context for a grid, $\mathbf{F} \in \mathbb{R}^{W' \times H' \times C}$ that is optimized for fast lookup and covers the entirety geographical area of interest. To build the grid, first, we calculate shapes for each cell in the grid using some fixed unit (e.g. 10×10m). Second, we build a spatial-index using the *R-Tree* algorithm [36], which allows us for any OSM shape to lookup all intersecting cells on average $O(\log_2(W'H'))$ time (see Alg. 1). As a consequence, all GPS points, $(lat_{u,i}, long_{u,i})$, that are located in same cell will be assigned the same geo-spatial context. This is an acceptable trade-off since $\log_2(W'H') \ll N$, and thus the data fusion cost can be considered linear on the number of GPS points.

For each of the subset of selected Open Street Map feature types to be captured, c, we find all intersecting cells, j, k, in $\mathbf{F}$, using the *R-Tree index*, and assign $\mathbf{F}_{j,k,c} = 1$, and otherwise 0 (see Alg. 2).

At this point any $\mathbf{I}_{u,i}$ is simply a subset of $\mathbf{F}$, specifically given that the corresponding GPS point $(lat_{u,i}, long_{u,i})$ is located in cell j, k then $\mathbf{I}_{u,i}$ can be calculated in constant time using (2).

$$j, k | (lat_{u,i}, long_{u,i}) \models \mathbf{I}_{u,i} = \mathbf{F}_{j-\frac{W}{2}...j+\frac{W}{2}, k-\frac{H}{2}...k+\frac{H}{2}} \tag{2}$$

A convenient visualization of $\mathbf{F}$ is presented in Figure 1. Intuitively, we can think of tensor $\mathbf{F}$ as C images having size $W' \times H'$, stacked one on top of another. Each image, $c \in \{1, \ldots, C\}$ corresponds to the c^{th} spatial-context feature, which we visualize with a distinct color on pixels where $\mathbf{F}_{j,k,c} = 1$, and transparent otherwise.

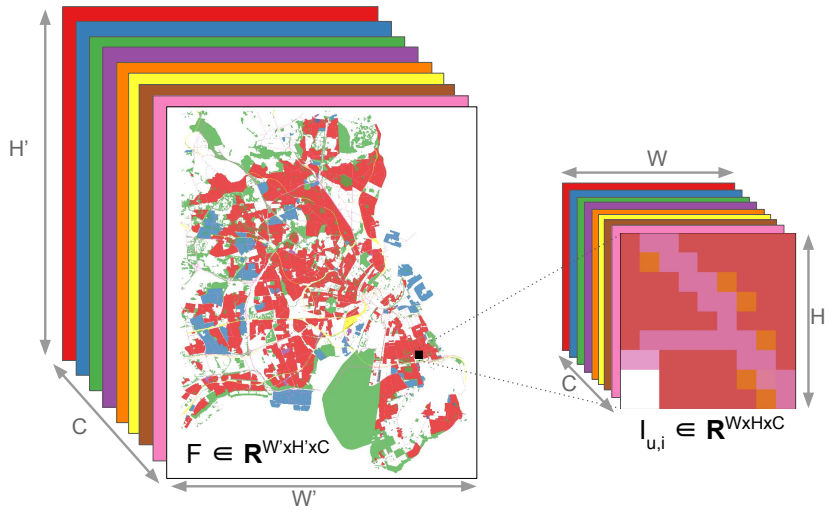

Figure 1: Visual representation of 8 of the 11 channels of the grid: landuse residential, landuse industrial, landuse meadow, landuse commercial, shops, rail roads, roads, bus stops.

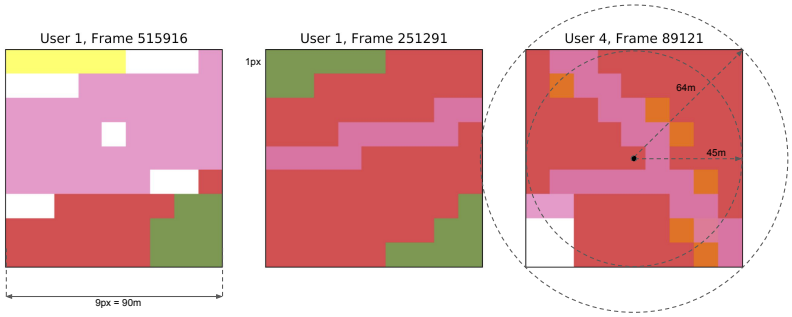

Figure 2: Example of the 9x9 frames used as input for the CNN model. The channels are visualized using the same colors as used in Fig. 1.

3.3. ANN architecture design and optimization

For the point-based binary classification of GPS trajectories, this paper investigates two different models, using different ANN methods, and two competing data representations, which are:

1. RNN, using only the kinematic features derived from GPS, as defined in (3);

2. RNN, using the representation with dummy variables defined in (6);

3. A combination of RNN and CNN, using the data representation defined in (7).

Against the method we propose in Point 3, we include also a rule-based baseline model and a random forest (see Sec. 3.4). We tested the former with GPS as input, and the latter with both input representations described in (3), and (6).

3.3.1. Structural hyperparameters

As we are performing a classification task, the cross entropy is our loss function[1], which we can optimize testing various strategies, introducing further hyperparameters, i.e. optimizer, regularization rate, learning rate, and number of epochs [37].

To cope with severe class imbalance, while minimizing the loss function we can penalize the *stop* in favor of *motion*, according to the relative weight normalized on the size of the smaller class. We can also avoid training and back-propagating on batches representing the larger class only, by setting a large value for the hyperparameter *batch size*.

To keep exploding gradients under control when gradient convergence becomes very unlikely for large noise variance [38], and avoid quick deterioration of the loss function, after back propagation we can apply a gradient clipping [39] strategy, introducing a specific hyperparameter named clipping rate.

To train a ANN model and specify its optimal parameters based on a training dataset, we pick one set containing one value for each of the hyperparameters listed in both this and the following sections, and we run one optimization loop.

To cope with the strong class imbalance mentioned in Sec. 1, which after training, on the validation set, translates into the classification of all observations as *stop*, we set a large batch-size and we penalize the larger class in the optimizer.

[1]Cross Entropy. Retrieved from web 26/11/2019.

Based on the results of each training cycle, we can assess whether and how to tune each HP within a new set of HP. Every new set must be tested in a new training cycle. In Tab. 1 we report both hyperparameters and corresponding range of experimented values, while in the following sections we present in depth the hyperparameters describing the final configuration of our three ANN models.

Other critical hyperparameters shared across different ANN structures, are the following.

1. To compound the output of complex configurations, e.g. CNN and RNN, we can use FFN towards the end, where we need to find the optimal number of layers and neurons per layer.

2. To capture non-linear relationships we can rely on different Activation Functions (AF) through the whole network.

3. To improve the learning performance, we can implement batch normalization [40] between each layer, beginning from the input, throughout the network output.

4. Softmax is a special AF, implemented in the last layer to output probabilities summing to 1[2].

5. To contain over-fitting, we can implement dropout [41] layers within the network architecture, and L2 regularization [37] in the optimization algorithm.

3.3.2. RNN using only kinematic features

RNN repeatedly transform a sequence of inputs, where each input has the same dimensionality, R, and the length of the sequence, L is a hyperparameter. The output is a function of input and hidden state. The latter is updated based on the input vector and itself. The hidden state, then, is used to process the next input vector. The dimension of the hidden state is a hyperparameter. Multiple RNN layers can be stacked within the same network architecture, where the ouput of one layer becomes the input of the next: the number of layers is a hyperparameter.

[2]Softmax. Retrieved from web 26/11/2019.

Table 1: List of hyperparameters and values' range tested in various ANN configurations.

Recurrent Neural Network Layers	$\in [1, 4]$
Hidden State Dimension	$\in [1, 30]$
Sequence length	$\in [3, 60]$
Convolutional Neural Network Layers	$\in [1, 4]$
Filters	$\in [16, 256]$
Filter Kernel	$\in [(1, 1), (5, 5)]$
Padding	$\in [0, 1]$
Stride	$\in [1, 2]$
Max Pooling Kernel Size	$\in [2, 3]$
Max Pooling Stride	$= 2$
Fully connected Layers	$\in [1, 4]$
Fully Connected Units	$\in [10, 2048]$
Dropout	$\in [0.2, 0.8]$
Activation Function	$\in$ {Rectified Linear Unit (ReLU), Leaky ReLU, Hyperblic Tangent (Tanh)}
Optimizer	$\in$ {Adam, Stochastic Gradient Descent (SGD), RMSProp}
L2 regularization Weight Decay	$\in [10^{-12}, 10^{-6}]$
Learning Rate	$\in [10^{-5}, 10^{-1}]$ implementing LR decay
Epochs	$\in [1, 100]$
Gradient Clipping Rate	$\in [0.3, 0.5]$
Batch Size	$\in [8, 12000]$
Items Shuffling during training	$\in$ {Yes, No}

1. Sequence length, L: Here it represents the sequence length in term of GPS points, thus time steps, that we feed into the network to preserve the temporal context of our time series.

2. Hidden State Dimension: Here we *store* previous sequences of GPS points, keeping a memory of the temporal context through the whole sequence-processing.

3. RNN Layers Number: By processing the time series in input, each layer outputs a new time series for the next layer, opening the possibility of decomposing and learning complex patterns. In case of bi-directional RNN, we have two blocks of layers. One will process the features de-

rived from our GPS time series forward, as $F(GPS_0) \rightarrow F(GPS_t)$; the other, backward, as $F(GPS_t) \rightarrow F(GPS_0)$. Both will contribute to the same output.

Let $p_{u,i}$ be the representation of any GPS point through kinematic features, such that:

$$p_{u,i} \models d_{u,i} \parallel br_{u,i} \parallel \mathbf{ToD}_{u,i} \tag{3}$$

In this configuration (see architecture in Fig. 3), we represents $p_{u,i}$ as in (3).

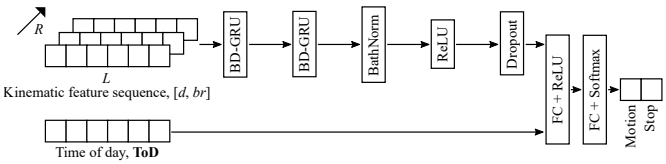

Figure 3: Network architecture for RNN using only kinematic features.

The kinematic features consists of distance, d, and bearing, br, i.e. $R = 2$. We arrange the features into a sequence with length, L and process through a RNN followed with batch normalization, ReLU activation and dropout. The output of the RNN-block is concatenated with **ToD**, a dummy variable representing *time of day* [42]. We processed the resulting feature vector through a FFN.

In particular, to preserve the time dependency of the time-series, the RNN was a GRU unit [43]. As in [12], we took advantage of the motion laws behind the GPS trajectories, which should be confirmed despite the processing direction of the time series, and we configured GRU as bidirectional. With the bidirectional GRU, each element of the sequence receives the information, through the hidden state, from all the other elements. In this model, we classify the last element of the sequence only [42].

The final version of the model has about 60 000 parameters in total. The resulting optimal HP, selected with *TR* and *VA* partitions, are available in Tab. 2 and 3. In Tab. 8, we report the resulting performance estimated using the optimal set of HP to classify the *TE* partition.

Table 2: RNN architecture hyperparameters, final configuration for both RNN with GPS only and RNN with GIS+GIS fusion with dummy variables.

Recurrent Neural Network Layers	2
Hidden State Dimension	4
Fully connected Layers	2
Fully Connected Units	Layer 1→ 512
	Layer 2→ 100
Dropout	0.45
Activation Function	Rectified Linear Unit (ReLU)

Table 3: RNN optimization hyperparameters, final configuration for GPS only.

Optimizer	Adam
L2 regularization Weight Decay	10^{-10}
Learning Rate Decay	Epoch 0-40 → 0.1
	Epoch 41-50 → 0.01
Epochs	50
Batch Size	12 000
Gradient Clipping	0.5
Items Shuffling during training	Yes

Table 4: RNN optimization hyperparameters, final configuration for GPS+GIS with dummy variables.

Optimizer	Adam
L2 regularization Weight Decay	10^{-10}
Learning Rate Decay	Epoch 0-30 → 0.1
	Epoch 31-50 → 0.01
Epochs	50
Batch Size	13 500
Gradient Clipping	0.5
Items Shuffling during training	Yes

3.3.3. RNN using kinematic and geo-spatial features as dummy variables

From (2) we can obtain also a one-dimensional dummy variable representing the same spatial-context with lower resulution, which we define as:

$$\mathbf{D}_{u,i} \in \mathbb{R}^C \tag{4}$$

such that

$$\mathbf{D}_{u,i,c} = \begin{cases} 1, & \text{if } \Sigma_{j,k}\mathbf{I}_{u,i,j,k,c} > 0 \\ 0, & \text{if } \Sigma_{j,k}\mathbf{I}_{u,i,j,k,c} = 0 \end{cases} \tag{5}$$

The resulting alternative representation of each GPS position within the surrounding spatial context, using a dummy variable, is

$$p_{u,i} \models d_{u,i} \parallel br_{u,i} \parallel \mathbf{ToD}_{u,i} \parallel \mathbf{D}_{u,i} \tag{6}$$

This model is a twin of the model described in Sec. 3.3.2 and illustrated in Fig. 3. The difference is in the input $p_{u,i}$, which we model as described in (6), instead of (3). We include $\mathbf{D}$ with dimensionality C; thus, since we still include bearing and distance, $R = 2 + C$. This difference impacts on the total number of parameters of the model, in this configuration $\approx 82\,000$. Also the optimization HP are slightly different. The resulting HP are available in Tab. 2 and 4; the resulting performance, in Tab. 8.

3.3.4. CNN+RNN using kinematic and geo-spatial features as tensors

In CNN, the convolution of an input tensor is obtained by striding a filter over such a tensor. Convolutional layers can be stacked. The output of the previous layer is a tensor that becomes the input of the next layer. The output's size of each transformation, depends on the following hyperparameters: number of layers, number and size of filter kernels, filter strides, and padding strategies of the input.

1. Number of Filters. Each filter is responsible of learning a pattern present in the image, considering all its channels at the same time, but limited to the size of the filter kernel.

2. Filter Kernel Size. It is the portion of the image on which the filter transformation is applied. This transformation convolves the whole image.

3. Number of Layers. Multiple convolutional layers are responsible of learning more complex patterns, combining the simpler patterns learned by the previous layers.

4. Stride Size. Small strides provide redundancy in the convolutions, whereas large strides might limit redundancy.

5. Padding Size. Padding strategies, in combination with filter kernel size, allow control on the volume of the convolution's output, enabling deeper or shallower architectures.

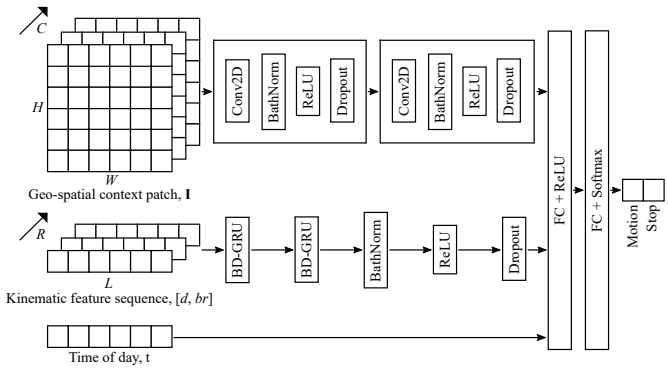

Figure 4: Network architecture for CNN+RNN model.

Between convolutional layers, we can have Max Pooling Layers. Intuitively, this transformation is very similar to a convolution, and is defined by two hyperparameters: filter kernel size, and stride. In this layer, a filter kernel strides across the output of the previous convolutional layer, but here it specializes in extracting the most relevant signals towards the next convolutional layer.

The representation of $p_{u,i}$ that we obtain by augmenting kinematic features with spatial context through (3) and (2), is the following:

$$p_{u,i} \models d_{u,i} \parallel br_{u,i} \parallel \mathbf{ToD}_{u,i} \parallel \mathbf{I}_{u,i} \tag{7}$$

Compared to a dummy variable describing C geo-spatial features, within a defined range centered in a GPS position, $\mathbf{I}_{u,i}$ provides more detail around the GPS position, augmenting the resolution from $\mathbb{R}^C$ to $\mathbb{R}^{W \times H \times C}$.

CNN are a perfect fit extract features from multi-dimension tensors as $\mathbf{I}_{u,i}$, which we use in this configuration augment the kinematic features derived from GPS, with spatial context information. Hence, we combined the network described in Sec. 3.3.2 with a CNN, as described in Fig. 4. As with the RNN configuration of Sec. 3.3.2 and 3.3.3, we classified only the last element of the sequence by assessing the hidden state. The configuration of this network required some further tuning. The resulting HP are available in Tab. 5 and 6; the resulting performance, in Tab. 8. The challenge here was to reduce the total amount of parameters, which in this configuration are approximately 100 000.

Table 5: CNN+RNN architecture hyperparameters, final configuration.

Recurrent Neural Network Layers	2
Hidden State Dimension	5
Convolutional Neural Network Layers	2
Filters	Layer 1→ 16
	Layer 2→ 8
Filter Kernel	3
Padding	1
Stride	1
Max Pooling Kernel Size	Layer 1→ 3
	Layer 2→ 2
Max Pooling Stride	1
Fully connected Layers	2
Fully Connected Units	Layer 1→ 512
	Layer 2→ 100
Dropout	0.45
Activation Function	Rectified Linear Unit (ReLU)

Table 6: CNN+RNN optimization hyperparameters, final configuration.

Optimizer	Adam
L2 regularization Weight Decay	10^{-10}
Learning Rate Inv.Decay	Epoch 0-10 → 0.01
	Epoch 11-50 → 0.1
Epochs	50
Gradient Clipping	0.3
Batch Size	13 500
Items Shuffling during training	Yes

3.4. Baselines grid-search

The extensive reviews provided in [8, 29, 30], report rule-based techniques as the most common methods for stop-detection and trip-segmentation. These methods perform classification based on, e.g., spatial-, time-, speed- or acceleration-thresholds. The same literature presents RF as one of the most effective supervised ML techniques. To assess the performance of our classifier, we pick two baselines: a rule-based method, which is unsupervised, and a RF, supervised.

1. Infostop [9], an efficient python package recently published on GitHub, allows unsupervised stop detection and labeling of stationary events from a GPS trajectory. By building a network that links stationary events, identified as nodes within a critical space-time range, and clus-

tering such a network using two-level Infomap[3], the algorithm provides a label for each point and stop event. This method does not support GPS+GIS data fusion in any way.

2. Sklearn, a very popular python package broadly used in Machine Learning, provides the RF algorithms used. To improve the classification accuracy and reduce over-fitting, RF relies on multiple decision tree predictors and averaging. To train this classifier, the first step is bootstrapping [45], which consist in sampling a number of training sub-sets from the main training dataset. In the second step, each training subset is split into in-bag [45] (IB) and out-of-bag [45] (OOB), sized one-third and two-thirds of such a sub-set. Then, while the OOBs are left out, a decision tree is constructed on each IB, by sampling randomly the attributes to determine the decision split [45]. Results from each decision tree will be averaged, thereby providing the final classification. Because the OOB step, RF performance estimation during training is also unbiased [30].

   K-fold cross-validation, however, may still be appropriate to replicate the same conditions when comparing RF performance with other methods (see Sec. 2.7).

For the grid search, we refer to the random forest only, as Infostop is not a parametric method. We performed two grid-searches: For $p_{u,i}$ represented as in (3), and for $p_{u,i}$ represented as in (6), which means with and without dummy variables. For the task, we used *GridSearchCV* a dedicated functionality of Sklearn, where we specified $TR \cup VA$ as training partition. After 5-fold estimations on the set of hyperparameters described in Tab. 7, we obtained two set of optimal HP, one for each data representation. Results show no significant difference between the two configurations (see Tab. 8).

Table 7: Random forest grid search hyperparameters.

Number of estimators	$\in \{100, 200, 500\}$
Max features	$\in \{\text{auto}, \text{sqrt}, \text{log2}\}$
Max depth	$\in \{4, 6, 7, 8\}$
Criterion	$\in \{\text{gini}, \text{entropy}\}$

[3]Infomap is a network clustering algorithm based on the Map equation [44].

4. ANN classification performance and benchmark

The first remark is about the class imbalance between stop and motion. Most of the public datasets, as for example [29], and [46], have relatively balanced classes and the stop class represent approximately 20% of the total. Unsurprisingly, the dataset we use presents realistic stop and motion proportions which are 80% and 20% of the total. Thus, the challenge for a model specialized on stop detection, on a realistic dataset, is detecting motion points. Suppose a model predicts the stop class only, as this large class overwhelms the model. Since stop is our target class, precision would be 80%, recall would be 100%, accuracy would be 80%, and F1-score would be 88.89%. By switching target class from stop to motion, the same prediction event would result in 0% precision, 0% recall, 0% accuracy and 0% F1-score (see Sec.2.3). Due to this heavy class imbalance, F1-weighted-average on the class size would be a misleading metric. Therefore, within the TE partition, to assess and compare our models we need to measure F1-average. In case of k-fold validation, the average of the k resulting F1-averages should be weighted on the size of each TE_{k-fold}. We used Python, Sklearn and PyTorch [47] for both models implementation and performance calculation.

4.1. Case study dataset

In 2018, The Center of Transport Analytics at the Technical University of Denmark, tested Mobile Market Monitor[4] (MMM), which is the commercial version developed from FMS, and collected GPS trajectories of $U = 12$ users, for about 24 user $\cdot$ days.

To manage the data fusion between GPS and GIS (see Sec. 2.5), we restricted the case study to the Copenhagen Capital area (see Fig. 1), consisting of approx. 1.5 million GPS trajectory points. Furthermore, we applied some preliminary cleansing, by excluding any point at the end of time intervals > 300 s and space intervals covered at speeds > 42 m/s ($\approx$150 km/h). The resulting dataset counts about 1.45 million GPS points [42].

We applied various filters; for example, removing any point at the end of time intervals > 60s (instead of > 300s), would further reduce the dataset by about 1.6% of the total. By removing also sequences of GPS observations counting less than 5 consecutive points, we obtain a clean dataset of over

[4]Mobile Market Monitor. Retrieved from web 26/11/2019.

$N \approx 1.42$ million GPS points. These filters are extremely effective in removing faulty GPS observations: speed above the threshold is unrealistic; time intervals between observations above 60s, are the symptom of, e.g., battery saving routines, or GPS satellites out of sight; segments with less than 5 consecutive observations represent time-series too short to be classified, and can be a symptom of noise in the data.

For $\mathbf{F}$, we chose a 1×1 cell size of 10×10m, and the resulting grid size is $H' = 2\,845$, $W' = 2\,331$, and thus grid consists of more than 6.6 million cells (see Fig. 1).

In our experiments, $\mathbf{I}_{u,i}$ represents a squared area of 90×90m using a cell size of 10×10m, and thus $H = W = 9$ (see Fig. 1). We capture up to $C = 11$ feature types from OSM data, for usage of the same cell space, such as bus stop, road, commercial land use, etc. Each of the 11 tensors' channels, is dedicated to one and only one spatial-context feature, which enables the representation of up to 11 mixed land-uses with 10×10m resolution.

Among the geo-spatial contexts relevant for transport choice and behavioral study [15, 48], we pick those that are represented in the available GIS. The selection consists of the following 11 features from OSM:

1. **Landuse residential**. A polygon that indicates that the area is primarily used for residential houses and homes. Knowing this information could help in detecting stops when a user is, e.g., at home or paying social visits.

2. **Landuse industrial**. A polygon that indicates that the area is primarily used for industrial buildings. The information could help in detecting stops when a user is, e.g., at work or on a business meeting.

3. **Landuse meadow**. A polygon that indicates that the area is primarily used for parks and forests. The information could help in detecting stops or walking move when a user is, e.g., doing sport or recreation activities.

4. **Landuse commercial**. A polygon around commercial areas, which could contribute in classifying, e.g., stops meal/eating brakes or entertainment.

5. **Shops**. POIs indicating, e.g., shops and restaurants, which could contribute in detecting stops for, e.g., eating out, shopping, pick-up or drop-off someone.

6-9 **Rail, Metro, Stations and Bus stops**. Points that could contribute in detecting, e.g., motion with specific transport modes, instant-stops for mode transfer, or long-discontinuities at intermediate stations.

10 **Major road network**. Segments corresponding to main road networks, which could help detecting motion by car, bus or bike.

11 **Traffic lights**. Points identifying nodes where a traffic light is present, which could contribute in detecting long-discontinuities of motion.

Fig. 1 shows a visual representation combining 8 of the 11 channels of the grid, where we overlay each channel with its own color and some transparency. Our classifier does not process this visual representation, but the tensor $\mathbf{I}$ defined in (2).

4.2. Dataset partitions for grid-search and run time

To prevent information spillover during the grid-search, we cannot build training, validation and test data partitions by blindly shuffling our ≈ 1.4 million GPS points, and then sampling randomly the *TR*, *VA*, and *TE* in some ideal proportions. We need to keep intact the sequence of the u^{th} time series, such that $\forall u \in \{1, ..., 12\}$, the partitions $GPS_u = \{1, ..., N_u\}$ represent continuous sequences on the time dimension, and are disjoint. Therefore, to perform the HP grid-search following the leave-one-out validation criterion described in Sec. 2.3, faster than a k-fold validation, first we sampled without replacement 9 random users out of 12, where we denote with s_n the user sampled on the n^{th} draw. Thus, we composed the partitions as in (8), (9),and (10) [42].

$$TR = \cup_{u=s_1}^{s_8} GPS_u = \{1, \ldots, N_{s_1} | \ldots | 1, \ldots, N_{s_n} | \ldots | 1, \ldots, N_{s_8}\}, \quad \textbf{\textit{card(TR) = 1,147,396}} \tag{8}$$

$$VA = GPS_{s_9} = \{1, \ldots, N_{s_9}\}, \quad \textbf{\textit{card(VA)=115,564}} \tag{9}$$

$$TE = \cup_{u=s_{10}}^{s_{12}} GPS_u = \{1, \ldots, N_{s_{10}} | 1, \ldots, N_{s_{11}} | 1, \ldots, N_{s_{12}}\}, \quad \textbf{\textit{card(TE) = 165,342}} \tag{10}$$

We used $TR \cup VA$ to find optimal hyperparameters, and TE to provide an estimation of the models' performance.

The grid-search performed for both methods and baselines (see Sec. 3.3, 3.4), using these partitions, results in the following computation time.

For ANN, we measured that each training cycle lasts ≈ 15 seconds per Epoch. Therefore, with training cycles composed by number of epochs $\in [20, 50]$, a grid-search requires a time interval $\in [0.3, 7.5] \cdot 10^3$ seconds per HP. One dedicated Graphic Processing Unit (GPU) ran all the computations.

For RF, the total time required for the grid-search is in the interval $[1.5\cdot, 2.1] \cdot 10^4$ seconds. In average, the time required for each hyperparameter is in the range $[1.1 - 1.6] \cdot 10^3$ seconds, with parallel computations across a 16 cores / 32 threads CPU.

As Infostop is not a parametric method, training time is null.

4.3. Benchmark

In the previous sections we described the process of finding optimal parameters for both ANN methods and baselines, using different representations of $p_{u,i}$. To find the best hyperparameters we used TR (8) and VA partitions (9), then we estimated the overall performance on the TE partition (10). The results are available in Tab. 8. To provide a distribution for performance and error, and allow a meaningful comparison across proposed methods and baselines, we fix the aforementioned optimal hyperparameters (see Sec. 3.3), and then we perform a 12-fold cross validation, split by users.

In this last step $VA = \varnothing$; TR and TE partitions have the same proprieties mentioned in Sec. 4.2, and are defined in (11).

$$TE_s = GPS_s \text{ and } TR_s = TE_s^{\complement}, \forall s \in [s_1, s_{12}] \tag{11}$$

First, we shuffled the list of users. Following the shuffled users' order, fold by fold, we rotated each available user in the TE partition, while all the rest of the users composed the TR partition. In this way, we can evaluate the error distribution user by user.

To check for linear correlations between method performance and data noise we define the noise as:

$$\text{Noise}_u = \frac{card(GPS_u^{\text{raw}}) - card(GPS_u^{\text{clean}})}{card(GPS_u^{\text{clean}})} \tag{12}$$

This coefficient is simply the percentage of points removed on each users' trajectory because of data cleansing.

We define three key performance indexes (KPIs):

- The correlation among F1-average$_{\text{u-fold}}$ and Noise$_u$ (12);
- F1-score, defined as the average of F1-average$_{\text{u-fold}}$, weighted on $card(TE_{\text{u-fold}})$, across the 12-folds;
- F1-std, defined as the average of F1-std$_{\text{u-fold}}$, weighted on $card(TE_{\text{u-fold}})$, across the 12-folds.

In addition to these KPIs, we look at precision and recall. $card(\cup_{u=1}^{12} TE_u) \approx 1.42 \cdot 10^6$ observations contribute to the estimation. The results are presented in Fig. 5 and 6. CNN+RNN with GPS+GIS data fusion (see Sec. 3.3.4) is the best performer on both F1-related KPIs, and second best after Infostop on linear correlation with noise. The ranking is consistent to the results validated out of sample, after the grid search (see Tab. 8). From Fig. 6 is evident that this model is significantly better in terms of recall of the motion class, and is less depended than RF on the GPS noise.

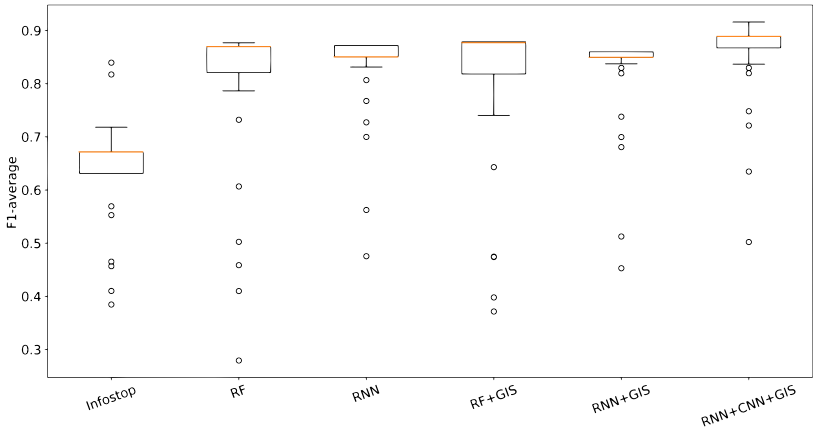

Figure 5: Box-plot of F1-score performance, across models and baselines, obtained with 12-fold cross validation with $N \approx 1.42$ million GPS observations in total.

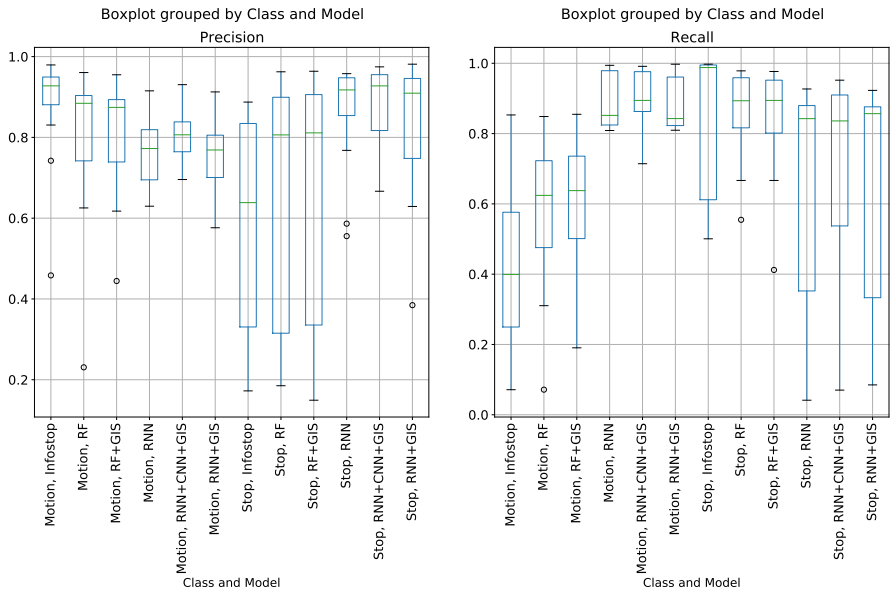

Figure 6: Distribution of precision and recall performance, across models and baselines, obtained with 12-fold cross validation with $N \approx 1.42$ million GPS observations in total.

4.4. Discussion

The previous sections show that the best classifier is the RNN+CNN model we propose. This model, specializes in the classification of the novel representation of $p_{u,i}$, as described in (7). Fig. 7 shows a strong negative linear correlation between the performance on each fold and the noise in the corresponding *TE* partitions defined in (12). Thus, when the tested partition presents high levels of Noise_u, F1-score is low, as expected. Our method is not significantly better when Noise_u is low. In contrast, when GPS observations are very noisy our method performs significantly better. The comparison with the other ANN models, which differ mainly on how the data is represented, show that the representation of (7) is quite robust to GPS noise. This performance come to the cost of heavier computations and more complex training, typical of ANN, which should be considered carefully before deployment on a larger scale (see Sec. 3.3 and 4.2).

Table 8: Comparison between ANN models and baselines. Performance off the sample, after grid search, computed on TE partition (see Sec. 4.2).

Model	Mode	Precision	Recall	F1-average
RNN with GPS only	Motion	66%	86%	82%
	Stop	95%	85%	
RNN with GPS+GIS fusion (6)	Motion	68%	86%	83%
	Stop	95%	86%	
CNN+RNN with GPS+GIS fusion	Motion	**81%**	**88%**	**89%**
	Stop	**96%**	**93%**	
Infostop [9] with GPS only	Motion	74%	73%	83%
	Stop	91%	91%	
Random Forest (RF) with GPS only	Motion	79%	80%	86%
	Stop	93%	93%	
RF with GPS+GIS fusion (6)	Motion	80%	79%	86%
	Stop	93%	93%	

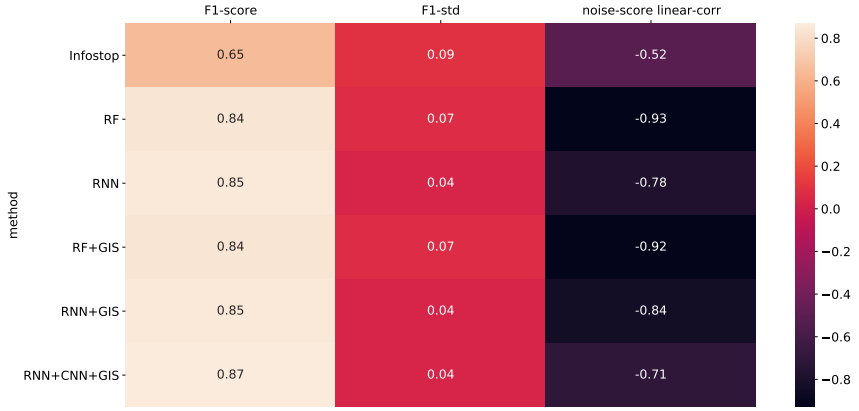

Figure 7: Methods benchmark. The correlation is among F1-average$_{\text{u-fold}}$ and Noise$_u$ (12). F1-score and F1-std, are the average of F1-average$_{\text{u-fold}}$ and F1-std$_{\text{u-fold}}$, weighted on $card(TE_{\text{u-fold}})$, across the 12-folds.

5. Conclusion

In this work, we used ANN to perform binary point-based classification - stop and motion - on $N \approx 1.42$ million points of GPS trajectories generated and validated by 12 users, during a test conducted with a smartphone-based travel survey (see Sec. 4.1).

We proposed the following models: RNN with GPS only data, RNN with

GPS+GIS data fusion as dummy variables, and a combination of CNN and RNN with GPS+GIS data fusion represented as a multi-dimensional tensor (see Sec. 3.3). We compared the performance against a clustering method, and a random forest. The latter, with and without GPS+GIS fusion represented as dummy variables. The results show that CNN+RNN is the best performer in terms of F1-average, 3% over both random forest configurations, and 6.5% over Infostop (see Tab. 8). Results are leave-one-out validated according to (8), (9), and (10).

To ease comparison among methods, performance and error distribution are estimated through a 12-fold cross validation with fixed optimal hyperparameters (11). The ranking is consistent with the aforementioned results. CNN+RNN classifier has the highest F1-score and the lowest F1-std (see Fig. 5). In particular, this method performs significantly better on recall for motion class (see Fig. 6), and when data are noisy (see Fig. 7).

ANN ran mostly on GPU; RF and Infostop, on CPU only (see Sec. 4). Although the process of GPS fusion with GIS, and training of ANN, is expensive in terms of CPU/GPU run time, in our experiments the difference seems handleable. Grid search and training time difference, seems mostly due to the larger number of hyperparameters of ANN. Yet, RF is faster of approximately one magnitude; Infostop does not require grid search at all. In contrast with RF, however, the method we propose for GPS+GIS fusion has a potential not exploited yet. On the one hand, this method could assist ANN configurations specializing in unsupervised learning, for the stop detection task. On the other hand, it could support multi-task classification of both transport-mode and trip-purpose, certainly supervised, possibly unsupervised. Future research will verify whether this potential can translate into a tangible asset for SBTS.

6. Competing interests

The authors declare that they have no competing interests.

References

[1] V. Servizi, F. C. Pereira, M. K. Anderson, O. A. Nielsen, Mining user behaviour from smartphone data: a literature review, 2019. arXiv:1912.11259.

[2] M. Bierlaire, J. Chen, J. Newman, A probabilistic map matching method for smartphone GPS data, Transportation Research Part C: Emerging Technologies 26 (2013) 78–98. URL: http://dx.doi.org/10.1016/j.trc.2012.08.001. doi:10.1016/j.trc.2012.08.001.

[3] Y. Kim, F. C. Pereira, P. C. Zegras, M. Ben-akiva, Activity Recognition for a Smartphone and Web- Based Human Mobility Sensing System, IEEE Intelligent Systems 33 (2018) 5–23. doi:10.1109/MIS.2018.043741317.

[4] S. A. H. Zahabi, A. Ajzachi, Z. Patterson, Transit trip itinerary inference with gtfs and smartphone data, Transportation Research Record 2652 (2017) 59–69. URL: https://doi.org/10.3141/2652-07. doi:10.3141/2652-07. arXiv:https://doi.org/10.3141/2652-07.

[5] H. Christiansen, Key figures from the Danish national travel survey, 2006. URL: https://www.cta.man.dtu.dk/english/national-travel-survey.

[6] A. N. Koushik, M. Manoj, N. Nezamuddin, Machine learning applications in activity-travel behaviour research: a review, Transport Reviews 0 (2020) 1–24. URL: https://doi.org/10.1080/01441647.2019.1704307. doi:10.1080/01441647.2019.1704307. arXiv:https://doi.org/10.1080/01441647.2019.1704307.

[7] A. C. Prelipcean, G. Gidofalvi, Y. O. Susilo, Measures of transport mode segmentation of trajectories, International Journal of Geographical Information Science 30 (2016) 1763–1784. URL: http://dx.doi.org/10.1080/13658816.2015.1137297. doi:10.1080/13658816.2015.1137297. arXiv:arXiv:1505.06786v1.

[8] L. Shen, P. R. Stopher, Review of GPS Travel Survey and GPS Data-Processing Methods, 2014. doi:10.1080/01441647.2014.903530.

[9] U. Aslak, Python package for detecting stop locations in mobility data, 2019. URL: https://github.com/ulfaslak/infostop.

[10] S. Dabiri, K. Heaslip, Inferring transportation modes from GPS trajectories using a convolutional neural network, Transportation Research Part C: Emerging Technologies 86 (2018) 360–371. URL: https://doi.org/10.1016/j.trc.2017.11.021. doi:10.1016/j.trc.2017.11.021.

[11] G. Xiao, Z. Juan, C. Zhang, Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization, Transportation Research Part C: Emerging Technologies 71 (2016) 447–463. URL: http://dx.doi.org/10.1016/j.trc.2016.08.008. doi:10.1016/j.trc.2016.08.008.

[12] X. Jiang, E. N. de Souza, A. Pesaranghader, B. Hu, D. L. Silver, S. Matwin, TrajectoryNet: An Embedded GPS Trajectory Representation for Point-based Classification Using Recurrent Neural Networks, in: Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering, 2017, pp. 192–200. URL: http://arxiv.org/abs/1705.02636. arXiv:1705.02636.

[13] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. De Macedo, B. Moelans, A. Vaisman, A model for enriching trajectories with semantic geographical information, in: GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, 2007. doi:10.1145/1341012.1341041.

[14] P. Stopher, Q. Jiang, C. FitzGerald, Processing GPS data from travel surveys, in: 28th Australasian Transport Research Forum, ATRF 05, 2005.

[15] N. Schuessler, K. W. Axhausen, Processing raw data from global positioning systems without additional information, Transportation Research Record 2105 (2009) 28–36. URL: https://doi.org/10.3141/2105-04. doi:10.3141/2105-04.

[16] A. Tietbohl, V. Bogorny, B. Kuijpers, L. O. Alvares, A clustering-based approach for discovering interesting places in trajectories, in: Proceedings of the ACM Symposium on Applied Computing, 2008. doi:10.1145/1363686.1363886.

[17] Y. Zheng, L. Zhang, X. Xie, W. Y. Ma, Mining interesting locations and travel sequences from GPS trajectories, in: WWW'09 - Proceedings of the 18th International World Wide Web Conference, 2009. doi:10.1145/1526709.1526816.

[18] R. Guidotti, R. Trasarti, M. Nanni, TOSCA: TwO-Steps Clustering Algorithm for personal locations detection, in: GIS: Proceedings of the

ACM International Symposium on Advances in Geographic Information Systems, 2015. doi:10.1145/2820783.2820818.

[19] L. Xiang, M. Gao, T. Wu, Extracting stops from noisy trajectories: A sequence oriented clustering approach, ISPRS International Journal of Geo-Information (2016). doi:10.3390/ijgi5030029.

[20] Y. Wang, D. McArthur, Enhancing data privacy with semantic trajectories: A raster-based framework for gps stop/move management, Transactions in GIS 22 (2018) 975–990. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12334. doi:10.1111/tgis.12334. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/tgis.1

[21] B. Thierry, B. Chaix, Y. Kestens, Detecting activity locations from raw GPS data: A novel kernel-based algorithm, International Journal of Health Geographics (2013). doi:10.1186/1476-072X-12-14.

[22] R. Hariharan, K. Toyama, Project lachesis: Parsing and modeling location histories, in: M. J. Egenhofer, C. Freksa, H. J. Miller (Eds.), Geographic Information Science, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 106–124.

[23] P. Nurmi, J. Koolwaaij, Identifying meaningful locations, in: 2006 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous, 2006. doi:10.1109/MOBIQ.2006.340429.

[24] G. Xiao, Q. Cheng, C. Zhang, Detecting travel modes from smartphone-based travel surveys with continuous hidden Markov models, International Journal of Distributed Sensor Networks (2019). doi:10.1177/1550147719844156.

[25] L. Liao, D. Fox, H. Kautz, Extracting places and activities from GPS traces using hierarchical conditional random fields, International Journal of Robotics Research (2007). doi:10.1177/0278364907073775.

[26] L. Gong, H. Sato, T. Yamamoto, T. Miwa, T. Morikawa, Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines, Journal of Modern Transportation (2015). doi:10.1007/s40534-015-0079-x.

[27] F. Zhao, A. Ghorpade, F. C. Pereira, C. Zegras, M. Ben-Akiva, Stop detection in smartphone-based travel surveys, Transportation Research Procedia 11 (2015) 218–226. URL: http://dx.doi.org/10.1016/j.trpro.2015.12.019. doi:10.1016/j.trpro.2015.12.019.

[28] C. Zhou, H. Jia, Z. Juan, X. Fu, G. Xiao, A data-driven method for trip ends identification using large-scale smartphone-based gps tracking data, IEEE Transactions on Intelligent Transportation Systems 18 (2017) 2096–2110.

[29] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, D. Roggen, Enabling reproducible research in sensor-based transportation mode recognition with the sussex-huawei dataset, IEEE Access (2019). doi:10.1109/ACCESS.2019.2890793.

[30] A. Yazdizadeh, Z. Patterson, B. Farooq, An automated approach from gps traces to complete trip information, International Journal of Transportation Science and Technology 8 (2019) 82 – 100. URL: http://www.sciencedirect.com/science/article/pii/S2046043018300236. doi:https://doi.org/10.1016/j.ijtst.2018.08.003.

[31] S. Dabiri, C. Lu, K. Heaslip, C. K. Reddy, Semi-supervised deep learning approach for transportation mode identification using gps trajectory data, IEEE Transactions on Knowledge and Data Engineering 32 (2020) 1010–1023.

[32] K. Lim, X. Jiang, C. Yi, Deep clustering with variational autoencoder, IEEE Signal Processing Letters 27 (2020) 231–235.

[33] C. M. Bishop, Pattern Recogniton and Machine Learning, 2006.

[34] D. Li, T. Miwa, T. Morikawa, Modeling time-of-day car use behavior: A bayesian network approach, Transportation Research Part D: Transport and Environment 47 (2016) 54 – 66. URL: http://www.sciencedirect.com/science/article/pii/S1361920916302334. doi:https://doi.org/10.1016/j.trd.2016.04.011.

[35] OpenStreetMap contributors, Planet dump retrieved from https://planet.osm.org , 2017. URL: https://www.openstreetmap.org.

[36] A. Guttman, R-trees: A dynamic index structure for spatial searching, SIGMOD Rec. 14 (1984) 47–57. URL: http://doi.acm.org/10.1145/971697.602266. doi:10.1145/971697.602266.

[37] T. van Laarhoven, L2 regularization versus batch and weight normalization, CoRR abs/1706.05350 (2017). URL: http://arxiv.org/abs/1706.05350. arXiv:1706.05350.

[38] Y. Bengio, P. Y. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 2 (1994) 157–66.

[39] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: 30th International Conference on Machine Learning, ICML 2013, 2013. arXiv:1211.5063.

[40] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Arxiv (2015) 1–11. doi:10.1007/s13398-014-0173-7.2. arXiv:1502.03167.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958. doi:10.1214/12-AOS1000. arXiv:1102.4807.

[42] V. Servizi, N. C. Petersen, Source code for Stop Detection for Smartphone-based travel surveys using ANN methods, 2019. https://github.com/niklascp/dtu-deep-learning-project.

[43] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014. doi:10.3115/v1/d14-1179. arXiv:1406.1078.

[44] M. Rosvall, D. Axelsson, C. T. Bergstrom, The map equation, European Physical Journal Special Topics 178 (2009) 13–23. doi:10.1140/epjst/e2010-01179-1. arXiv:0906.1405.

[45] L. Breiman, Manual on setting up, using, and understanding random forests v4.1, 2002. URL: https://www.cta.man.dtu.dk/english/national-travel-survey.

[46] Y. Zheng, H. Fu, Geolife GPS trajectory dataset - User Guide, Technical Report November 31, 2011. URL: http://research.microsoft.com/apps/pubs/?id=152176{%}5Cnhttp://research.microsoft.com/apps/pubs/default.aspx?id=152176.

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[48] I. Semanjski, S. Gautama, R. Ahas, F. Witlox, Spatial context mining approach for transport mode recognition from mobile sensed big data, Computers, Environment and Urban Systems 66 (2017) 38–52. doi:10.1016/j.compenvurbsys.2017.07.004.

Algorithm 1: Map WGS84-grid to Pixel-grid

Result: Reusable Pixel/WGS84 Map, saved on Disk
Input : Square coordinates on UTM32 grid
Output: Map of i^{th} pixel with i^{th} WGS84 cell

`/* initialize geographical area */`
$x_{min}, y_{min} \leftarrow$ lower-bound(SquareCoordinates)
$x_{max}, y_{max} \leftarrow$ upper-bound(SquareCoordinates)
`/* set resolution per pixel (m) */`
$dx \leftarrow dy \leftarrow 10$

`/* image size of spatial context */`
$W', H' \leftarrow \frac{x_{max}-x_{min}}{dx}, \frac{y_{max}-y_{min}}{dy}$
`/* initialize rtree index [36] */`
$idx \leftarrow$ init-rtree-index()
`/* initialize pixel count */`
$i \leftarrow 0$

foreach *x in range(0, W')* **do**
  foreach *y in range(0,H')* **do**
    $x_1 = x_{min} + x * dx$
    $y_1 = y_{min} + y * dy$
    $x_2 = x_{min} + (x + 1) * dx$
    $y_2 = y_{min} + (y + 1) * dy$
    `/* Set UTM32 bounds */`
    cellUTM32 = setSquare($A_{(x_1,y_1)}, B_{(x_1,y_2)}, C_{(x_2,y_2)}, D_{(x_2,y_1)}$)
    `/* Set WGS84 bounds */`
    cellWGS84 = transform(cellUTM32)
    `/* Map` i^{th} `pixel and` i^{th} `cell */`
    idx(i) = (cellWGS84, x, y)
    i=i+1
  end
end
CloseAndSave(idx)

Algorithm 2: Spatial context representation

Result: Image representation of geo-spatial context
Input : rtreeIndex (see Alg. 1), geo-spatial context features (on WGS84 grid)
Output: **F** of size $W' \times H' \times C$

```
/* load rtree index [36]                                    */
```
$idx \leftarrow$ load-rtree-index()
```
/* initialize empty tensor                                  */
```
$\mathbf{F}\,(W' \times H' \times C) \leftarrow 0$
```
/* initialize feature index                                 */
```
$C \leftarrow 0$

foreach *feature in SpatialContextFeatures* **do**

```
    /* load shapes on WGS84 grid                            */
```
    shapes = queryOpenStreetMap(feature)
```
    /* pick pixels intersecting shapes                      */
```
    pixels = intersect(idx, shapes)

    foreach *pixel in pixels* **do**
        F(x,y,C)=1
```
        /* each pixel includes coordinates x ∈ [0, W'] and
           y ∈ [0, H'] (see Alg. 1)                         */
```
    end
    C=C+1
end

4 Paper C: ``Is not the truth the truth?'': Analyzing the Impact of User Validations for Bus In/Out Detection in Smartphone-based Surveys

The following pages contain the article:

V. Servizi, D. R. Persson, F. C. Pereira, H. Villadsen, P. Bækgaard, I. Peled, and O. A. Nielsen (2021). ""Is not the truth the truth?": Analyzing the Impact of User Validations for Bus In/Out Detection in Smartphone-based Surveys". In: *IEEE ITS Magazine (UNDER REVIEW)*.

Please cite accordingly.

Part of this work was presented at the *"Annual Transport Conference at Aalborg University"*, Aalborg, August, 2021.

Servizi, V., Persson, D. R., Bækgaard, P., Villadsen, H., Peled, I., Rich, J., Pereira, F. C. and Nielsen, O. A. (2021, August). Context-Aware Sensing and Implicit Ground Truth Collection: Building a Foundation for Event Triggered Surveys on Autonomous Shuttles: Article. In *Proceedings from the Annual Transport Conference at Aalborg University (Vol. 28, No. 1)*.

"Is not the truth the truth?": Analyzing the Impact of User Validations for Bus In/Out Detection in Smartphone-based Surveys

Valentino Servizi[a,*], Dan R. Persson[b], Francisco C. Pereira[a], Hannah Villadsen[c], Per Bækgaard[b], Inon Peled[a], Otto A. Nielsen[a]

[a]*Department of Technology, Management and Economics*
Technical University of Denmark (DTU)
Kgs. Lyngby Denmark
[b]*Department of Applied Mathematics and Computer Science*
DTU
[c]*Department of People and Technology*
Roskilde University

Abstract

Passenger flow allows the study of users' behavior through the public network and assists in designing new facilities and services. This flow is observed through interactions between passengers and infrastructure. For this task, Bluetooth technology and smartphones represent the ideal solution. The latter component allows users' identification, authentication, and billing, while the former allows short-range implicit interactions, device-to-device. To assess the potential of such a use case, we need to verify how robust Bluetooth signal and related machine learning (ML) classifiers are against the noise of realistic contexts. Therefore, we model binary passenger states with respect to a public vehicle, where one can either be-in or be- out (BIBO). The BIBO label identifies a fundamental building block of continuously-valued passenger flow. This paper describes the Human-Computer interaction experimental setting in a semi-controlled environment, which involves: two autonomous vehicles operating on two routes, serving three bus stops and eighteen users, as well as a proprietary smartphone-Bluetooth sensing platform. The resulting dataset includes multiple sensors' measurements of the same event and two ground-truth levels, the first being validation by participants, the second by three video-cameras surveiling buses and track. We performed a Monte-Carlo simulation of labels-flip to emulate human errors in the labeling

process, as is known to happen in smartphone surveys; next we used such flipped labels for supervised training of ML classifiers. The impact of errors on model performance bias can be large. Results show ML tolerance to label flips caused by human or machine errors up to 30%.

Keywords: Ground-truth, D2D interactions, Autonomous vehicles, Bluetooth low energy, Internet of things

1. Introduction

Passenger flow is a fundamental component for capacity estimation of public transport and for designing adequate infrastructure and services [1]. On bus transport, this flow measures passengers' variations in time and space, on the vehicles [2]. Multiple approaches promise real-time passenger flow estimation, but even the most advanced ones struggle with imprecision due to counting passengers indirectly, such as when paying by cash, or traveling without ticket [2]. Although autonomous buses and the internet of things (IoT) offer the opportunity of exploiting D2D (device to device) interactions for passenger flow beyond ticketing [3], available solutions such as check-in/check-out (CICO), walk-in/walk-out(WIWO), or be-in/be-out (BIBO) [4, 5] all seem prone to errors. For example in the CICO case, using radio-frequency identification (RFID) technology for the interactions between smart-cards and readers, people often forget either the CI or CO action. In the WIWO case, multiple users can enter the same gate at the same time and confuse the counter. To contribute improving passengers' count accuracy and user experience in public transportation, we focus on enabling next generation BIBO for ticket-less trips. This application could allow passengers, for example, to pay with contact-less, radio-based identification, and communication via smartphone-Bluetooth without human intervention and without explicit interaction [6]. This approach has the added advantage of being the most user-friendly for the growing population of smartphone-users, which is above 40% worldwide and up to 80% in western countries [7], since it depends only on the user carrying his/her device as he/she would normally do.

*Corresponding author. Email: valse@dtu.dk

Although the global positioning system (GPS) is one of the most reliable and adopted technologies for outdoor tracking [8], GPS shows important limitations in urban areas [9]. The specific radio-signal frequency requires line of sight between sender and receiver, thus being affected by reflections from tall buildings and clouds. Similarly, Bluetooth is one of the principal technologies for proximity detection [10] applied to indoor tracking, and the specific radio-signal frequency brings other limitations. For example, a smartphone-based travel survey on the Silver Line bus rapid transit in Boston, Massachusetts, deployed BIBO technology [11], as a context detection system for a service quality survey, and so avoided collecting ground-truth, in form of labels, from surveyed passengers; D2D implicit interaction between smartphones and Bluetooth devices installed on buses verified passengers' presence aboard, independently from GPS sensors. In the same study, the authors expose cases where successful BIBO verification via GPS otherwise failed via Bluetooth, because smartphones could not receive Bluetooth signal within the bus, probably due to human body impedance relative to smartphone and Bluetooth device position. While a large body of literature presents a successful case for Bluetooth as indoor positioning technology, no previous work that we are aware of analyses in detail its use as independent measurement for labels and the impact labeling errors on the BIBO classifier performance. In this use case, Bluetooth reception errors might present themselves as flipping- and outlying-labels [12], negatively impacting ML training and magnifying misclassifications.

Flipping-labels are known as items that human or machine classifiers labeled with a wrong class, despite the true one existing in the dataset; outlying-labels are items that belong to none of the classes in the dataset, but were mistakenly labeled as one of these classes [13]. The impact of these two problems on ML classifiers is extensively studied for independent and identically distributed (IID) datasets [14, 13, 15, 16, 17, 12, 18, 19, 20, 21], such as for images, but not for time-series, such as Bluetooth or space-time GPS trajectories.

To bridge the gap between the limitations mentioned in the preceding two paragraphs, we conducted a case study on a BIBO, smartphone and Bluetooth Low Energy (BLE) based system, with the following research questions:

1. During the ground-truth collection, what is the users' response to wrong labels?

2. After ground-truth collection, what is the ML classification perfor-

mance based on various features extracted from BLE, GPS and accelerometer?

3. What is the resilience of ML supervised methods to flipping-labels?

Fig. 1 shows the process we executed through the following steps. First, we designed and implemented a smartphone-BLE platform. Second, we set up and ran an experiment involving a simple transport network composed of two autonomous buses operating on two routes and three bus stops, with a BLE device on each bus and bus stop. Third, we involved eighteen users and we video-recorded each of their trips through this network; simultaneously, users' smartphones native (Android and iOS) application programming interface (API) read BLE devices' signal strength and classified the transport mode from the time-series of the inertial navigation system (INS). Our proprietary application stored these trajectories on a database. Fourth, we labeled the trajectories using video recordings as ground-truth with BIBO binary labels. Fifth, we created a Monte Carlo (MC) process to simulate labeling errors, i.e. flipping-labels, with various noise levels on recorded trip time-series. Finally, to understand the error tolerance, we evaluated and compared multiple classifiers, both on true and noisy labels.

The experimental setting incorporates multiple real-world conditions typical of urban high density contexts: overlapping BLE fields, multiple makes and types of smartphones, native applications for the two main operating systems (OS), bus switching routes, bus moving at low speed, subjective preferences on how and where users carry their smartphones, or where they stand, both while traveling on bus and waiting at the bus stop. In such a BIBO system setup, we yield results suggesting that BLE signal alone is robust to labeling errors, and performs significantly better than commercially-available classifiers based on INS.

2. Related Work

This section focuses on two main bodies of literature that contribute to expose perspectives relevant for this use case: one on deployment of BLE beacons networks and signal processing for location prediction and activity classification, the other on the problem of label noise for ML classifiers. This section pinpoints candidate parameters and methods considered for designing the experimental setup of this work.

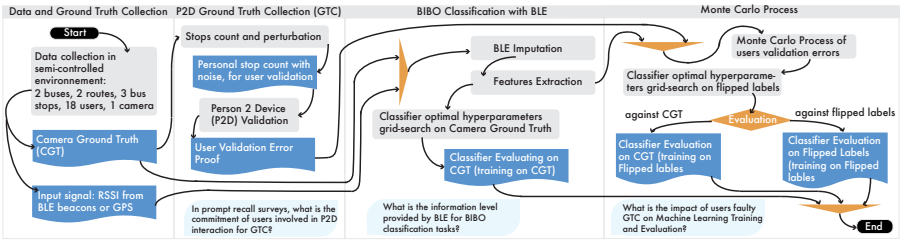

Figure 1: Experiment workflow.

2.1. Bluetooth applications

BLE stems from Bluetooth and WiFi protocols, and specializes in IoT applications; the communication is one-to-many, involves few bits of data to be broadcast frequently, and requires no pairing operation with other devices. All these properties make BLE technology particularly suitable for proximity detection [8]. Although in some field is heavily unbalanced towards other sensors, such as GPS and INS [8], BLE and WiFi are considered promising technologies even for transport mode detection in complex multimodal transport chains [22, 23, 24, 25, 26, 27]. For example, Bjerre-Nielsen et al. [27] perform transport mode detection based on received signal strength (RSSI) from Wi-Fi and Bluetooth signals, measured in decibels. The study analyzes and compares three supervised classifiers: random forest, logistic regression, and support vector machines. None of these methods involves artificial neural networks.

For implicit BIBO classification, Narzt et al. [6] propose an architecture where Bluetooth receivers are inside the bus while passengers carry a BLE device. The study carries out several experiments to recreate bus-space realistic conditions and analyses multiple configurations for Bluetooth receivers and device positions. No real users nor vehicles are involved in the study. However, the conclusion cautiously supports the hypothesis that larger-scale deployment of such a system is feasible. To further investigate potential interactions with the environment, the study highlights the need for a survey under realistic conditions from a larger-scale deployment perspective.

An independent, complementary, and substantial body of literature focuses on multiple sensors and algorithms for Mobile Anchor Node Assisted Localization [28], where WiFi and BLE signals are extensively studied in

general, and in particular for indoor tracking [29]. Among the methods available, geometric approaches are widespread, e.g., based on the Friis equation [30], and trilateration [31]. These methods rest on the knowledge of each device position and radio-signal propagation physics to approximate a receiver's location based on reception strength. Prevalent RSSI fingerprints approaches are ML-based, e.g., on k-nearest-neighbor and Kalman-Filters (KL) [32, 33, 34]. These algorithms rely on mapping a geo-spatial context with a sample of signal-strength-records, received from the devices on the range; grid resolution on the mapped space and signal-sample-size depend on the location accuracy required by the use case.

A natural extension of these technologies in the field of intelligent transport systems, is the study of vehicles to anything (V2X) communication. Whereas Bluetooth in general is not considered optimal for bi-directional communication due to slow paring process [35], BLE technology is substantially different and is able to trigger events in smartphones' OS, without any paring operation [36].

In summation, the above works indicate several pre-requisites for successful BLE application: (i) BLE signal transmission rate above 0.3 Hz; (ii) The density of the Bluetooth beacons network above one device every 30 square meters; (iii) Appropriate imputation of RSSI readings.

2.2. Noisy labels in machine learning classifiers

The problem of noisy data receives a lot of attention from the research community. The cause of noise in labels is manifold and use case dependent. For example, crowd-sourced labeling of images relies on expertise and attention of labelers, which they may not always have [19]. Similarly, in prompted recall surveys, users validate travel diaries with different dedication levels, and may therefore, negatively affect the quality of what is often perceived as ground-truth [8]. Consequently, noisy labels in turn negatively affect the classification accuracy of supervised or semi-supervised ML methods, which depend on these labels in the training process.

Previous systematic studies on noisy labels compare multiple supervised classifiers on multiple synthetic datasets [18], and analyze how robust learning algorithms are to noise [37, 38, 39]. Another research line works on noise cleansing or labels correction methods [40, 41, 42]. Numerous alternative approaches exist for improving classification accuracy in the presence of noisy labels, for example: (i) To pinpoint wrong labels, majority voting across multiple neural networks [43]. (ii) To learn labels' noise distribution, specialized

layers for artificial neural networks [19, 44, 45]. (iii) To predict the noise affecting training, conditional noise models [46]. (iv) To reduce the number of labels necessary for training, semi-supervised approach achieved with generative models [47]. (v) To leverage on existing high-quality sub-set of labels, propagation methods of these labels [48]. (vi) To learn labels on the fly and reduce human errors, graph-based label propagation methods [49].

However, despite the wide body of previous work, our use case did not receive enough attention, and thus no conclusions can be drawn as of its potential. Whereas in existing work datasets are exclusively IID, we investigate on dependent observations over time. Further, we explore the impact of varying labels' quality on model performance, presenting a real-world dataset with high-quality ground-truth, and leveraging Monte Carlo simulations for labels' variation study. In contrast, existing works rely on synthetic datasets, which on one hand offer reliable ground-truth, but on the other hand yield biased measurements compared to real data. In addition, we try to answer the following two questions: (i) What is the commitment of users involved in a person to device (P2D) explicit interaction for ground-truth collection? (ii) What is the information level that BLE provides for BIBO classification tasks with optimal ground-truth?

3. Methods and materials

To assess BIBO error tolerance under the experiment setup, first we need to understand how users collect faulty ground-truth, and then use this knowledge to derive a Monte Carlo process generating the same noise on the labels. Next, we can provide a broader analysis over the impact of noisy labels on Machine Learning training and evaluation steps, and mostly we can carry out this analysis on real trajectories. Fig. 1 describes the methodological process we adopted; Figure 2 presents the BIBO platform we designed, implemented and deployed for data collection.

3.1. Sensing platform

The smartphone sensing platform's main components are the front-end applications and back-end. The front-end is specific, or native, for Android and iOS. The apps contain the following features: data collection from on-board sensors and native APIs, such as users' transport activities classification; data transfer to the back-end from a local buffer that avoids data loss in case of external connectivity problems; and lastly, real-time tracking

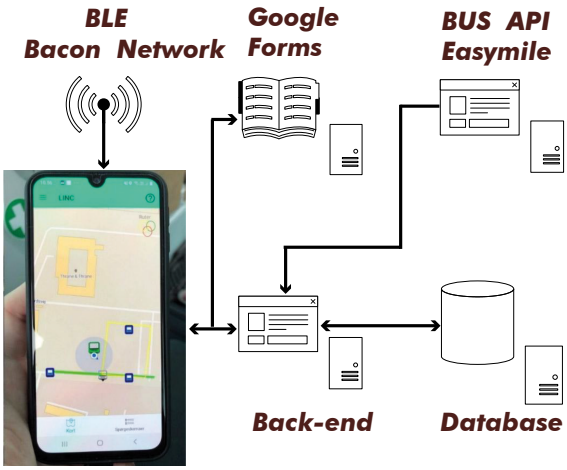

Figure 2: Sensing platform.

of buses on a map, with bus stops. The sensors we target are GPS, and BLE signal strength perceived from the BLE beacons network. To improve smartphones' battery efficiency, we monitor users' activities. We switch on and off smartphone GPS and data transfer, when the user is active and inactive, while the accelerometer detects the state. The back-end, which includes several specialized APIs, is responsible for exposing the information from the autonomous vehicles to the smartphone and handing the data from smartphones to the database.

As opposed to another architecture presented for BIBO [6], where users carry BLE devices, and buses are equipped with signal processing devices, we decide to follow the architecture of smartphone-based travel surveys [8], where users carry their smartphone, which is also a signal processing device, while buses are equipped with BLE devices. This configuration presents two main advantages. First, we extend the beacons' network outside the bus, at the bus stops. Second, we allow users to carry their phones and not interfere with their normal behavior while picking up signal from multiple sources. From the perspective of a larger-scale deployment, the experimental setup seems more realistic under these two conditions. The installation cost of a beacon device should be a fraction of the Raspberry-Pi deployed in [6]; furthermore, users should carry their smartphones only and not a new

device. This last element introduces a new random variable: The varying quality of sensors installed in different smartphone models, and the sample collected in this experiment is not representative of this broad population. Thus, to collect consistent data, we rely on the standardization of sensors and protocols represented on the aforementioned OS. We also collect the accelerometer-based activity recognition that these OS offer via APIs to discriminate between states, such as: automotive, bicycling, walking, running, stationary and unknown [50, 51].

3.2. Experiment setup and data

The possibility of replicating beacons' density of indoor settings, which should be $> \frac{1}{30\ m^2}$, is not realistic from this use case's scale-up perspective (see Sec. 2.1). However, installing devices both on the bus and on bus stops, which is more realistic, allows a temporary and nearly-optimal density of the beacons' network, at least between passengers' boarding and alighting, and when the bus stations in front of a bus stop.

The setup consisted of two autonomous vehicles operational on two distinct routes and three bus stops. To allow passengers' transfer between the buses, one bus-stop was shared between the routes, additionally sharing a segment of the test track; during the experiment, the buses' assignment to the route has been switched for technical problems, similar to real world settings. The BLE beacon network counted one device per bus and one device per bus stop, with five devices in total (see Fig. 2). Each device transmitted at the rate of 1.667 Hz and -8 dBm power. As these two settings affect the battery life of BLE beacons, the decision considers a realistic battery life expectation above one year, within the frequency recommendations from indoor studies (see Sec. 2.1).

Smartphone onboard sensors collected trajectories for twelve users only, for a total of $13,723$ *points*. We stored for each of these timestamps: GPS longitude and latitude; 5 RSSI readings from the BLE beacons network, one for each device, and transport-mode as classified by off-the-shelf accelerometer-based classifiers available on both Android and iOS operating system. The high number of unavailable trajectories, approximately $\frac{1}{3}$ of the total, is the result of two distinct problems. Four users did not grant the permission to access location sensors, resulting in no database records.

To count passengers' flow, we installed a high-resolution video-camera pointing to the buses' doors at each bus stop as the principal ground-truth. However, the three cameras in combination also allowed the full surveillance

of the track. A problem with the video-cameras, prevented the determination of high quality ground-truth for three users. Using video footage as ground-truth for the trajectories successfully collected from the remaining users, we provided a set of binary labels consistent with the BIBO model [6], on each point: inside or outside the bus.

To collect feedback from users after the experiment and link each user's feedback to the other data collected from smartphones, we rely on electronic forms with a pre-filled unique identifier corresponding to the user.

3.2.1. Procedure

We distributed a paper-based form to each user. The form included the experiment description and the information on data collection and use exclusive for research purposes (GDPR complainant). Then we briefed each participant on the following steps: (i) Install on smartphones the application we published to the application stores (for beta testing). (ii) Read general conditions and grant the application permission to access smartphone sensors and activity recognition. The latter performs transport mode detection [51, 50]. (iii) Wear a sleeve number to ease the ground-truth collection from video recordings. (iv) Use the transport network with the commitment to enter and exit the bus more than once, and with the possibility of walking between bus stops. (v) Count the total number stops, defined as the discontinuities between transportation modes, and expect a message stating our count, in the following days, with the request to validate or correct such a count.

Finally, we answered any questions raised by the participants, and from all the participants willing to participate we collected a paper-based signed authorization to proceed with experiment and data collection [1].

3.2.2. Participants

Active ground-truth collection P2D, which users provided in the days following the experiment, included fourteen valid replies. We counted the total number of stops for each user from video-recordings. To explore the users' commitment and the quality of a P2D ground-truth collection, we

[1] This project is a social science study, includes data and numbers only, is not a health science project, and does not include human biological material nor medical devices. Consequently, in Denmark, where the data collection took place, the Health Research Ethics Act provides a dispensation for notification to any research ethics committee.

introduced a level of noise in these counts before submitting the validation request, on a random sample of users. For the noise distribution, we assumed that validation errors could be Poisson distributed, similarly to OD matrix counts [52], as each error event is discrete and has minimal probability.

3.3. Wizard of Oz (WoZ) for P2D ground-truth collection

In this case, WoZ refers to the experimenter pretending that a BIBO system is operational on the test-bed [53]. The role of the user is to validate the measurements of such a BIBO system. Therefore, the user is briefed to count how many times he or she alighted from or mounted on a bus. To observe the P2D validation dynamic, the experimenter then provides WoZ's count to the user. In particular, as stated in Sec. 3.2, we assume that a Poisson distributed random error affects users' count. Since users' validation seems to support this hypothesis, we use this distribution to simulate users' validation errors within the Monte Carlo simulation described in Sec. 3.5, where counts errors propagate to the time-series' labels, flipping a BI in BO, or vice versa.

3.4. Data preparation and classifiers for BIBO

To assess BLE beacons' signal performance in determining users' presence inside or outside the buses, we use GPS as one of the benchmark. From GPS we extract the following features: distance between points, bearing, and speed [54, 55], which we process as time series extracting the same features extracted for BLE beacons. Table 4 presents the list of features collected in 10 seconds moving window. Further, from smartphones' OS we collect the binary classification automotive vs. everything else, compatible with BIBO in this context, which is based at least on accelerometer. Thus, we rely on the following tools, which we apply separately to BLE and GPS signal.

3.4.1. Framework

Scikit-learn is a popular python-based framework that includes several effective ML models. Random Forests (RF) represent a reliable and scalable supervised method for this task [56]. At the same time, Multi-layer perceptron (MLP) can be considered a building block of generative models, which can operate semi-supervised or unsupervised [47]. Therefore, we include these two supervised classifiers in the study. The following sections present further details, on preparation, training, and validation of the classifiers.

3.4.2. Random Forest

RF evolve from decision tree predictors, averaging results from multiple of these predictors. The effect is a more accurate classifier less prone to overfitting. The training phase starts with bootstrapping [57], which consists of several sub-samples with replacement from the training dataset. Each training sub-set is then split into in-bag [57] (IB) and out-of-bag [57] (OOB). The latter's size is one-third of such a sub-set, while the former accounts for the rest. A decision tree is constructed from each IB, while the attributes are sampled randomly to determine the decision split [57]. Finally, the RF output is aggregated over all individual trees, and the output is the class with the highest average probability, whereas in classical majority voting the output is the most common class prediction among trees.

3.4.3. Multi-layer perceptron

Perhaps we can consider it the most simple feed-forward artificial neural network [58]. MLP incorporates multiple layers for logistic regression. Multiple perceptrons, or neurons, compose each layer and handle nonlinearities through activation functions, such as sigmoids and rectified linear units (ReLU). For classification, each neuron's weight and bias is trained by minimizing the cross-entropy between the class predicted by the network and the ground-truth. These parameters are iteratively updated at each classification attempt, defined epoch, by back-propagating the resulting stochastic gradient towards the cross-entropy local minimum.

Training artificial neural networks requires large datasets. In this case, it is arguable whether the dataset size is appropriate or not. However, given the possible future extension of this experimental setup to real life operations, we are interested in investigating the potential.

3.4.4. Hyperparameters grid search

To perform this task we used *GridSearchCV*, a specialized library available in Sklearn. To obtain a set of optimal hyperparameters, we perform a 5-fold cross-validation on the training-set exploring those that Table 1 and Table 2 describe for RF and MLP. In a following step we train the classifier on the training-set, fixing these optimal hyperparameters, and we perform the evaluation on the test-set. Sec. 3.5 provides further details on this process within the simulation of ground-truth collection errors causing flipping-labels.

Table 1: Random forest hyperparameters search space.

Number of estimators	$\in \{10, 20, 100, 200, 500\}$
Max features	$\in \{\text{auto}, \text{sqrt}, \text{log2}\}$
Max depth	$\in \{3, 4, 6, 7, 8\}$
Criterion	$\in \{\text{gini}, \text{entropy}\}$

Table 2: Multi layer perceptron hyperparameters search space.

Hidden layers/sizes	$\in \{1 \text{ layer } (L) \rightarrow [50 \text{ neurons } (N)],$
	$3L \rightarrow [10N, 50N, 10N],$
	$4L \rightarrow [10N, 50N, 50N, 10N]\}$
Learning rate strategy	$\in \{\text{constant}, \text{invscaling}\}$
Learning rate coefficient	$\in \{10^{-2}, 10^{-3}\}$
Activation funcions	$\in \{\text{ReLU}\}$
Optimizer	$\in \{\text{adam}\}$

3.4.5. Validation process

The risk of information spill-over between training- and validation-set is higher when working with time-series. [59] shows that the violation of the out-of-sample (OOS) principle is not rare in the existing literature. Such a violation yields a virtual higher performance when evaluating a classifier, resulting in a biased measurement. Even in the assumption of non-violation of the OOS principle, researchers have several options for assessing a classifier, such as hold-out, leave-one-out, and cross-validation. (i) In the hold-out case, typically, the training-set should use approximately $\frac{2}{3}$ of the dataset; the validation-set, the remaining $\frac{1}{3}$. Training and validation proceed only once and yield the model performance based on the sole validation-set. (ii) In the leave-one-out case, the training-set should use a dataset's random sample of size $M - 1$, where M is the dataset's cardinality; the validation-set, the remaining one sample. Training and validation proceed M times and yield the model performance as a distribution over M-validations. (iii) In the cross–validation case, the dataset is split into N equal partitions; the training-set uses N-1 partitions, while the validation-set uses the remaining one partition. Training and validation proceed N times and yield the model performance as a distribution over N-validations. The approach (i) is computationally light-weight, but the resulting performance estimation might be negatively biased; (ii) is unbiased but could present a large performance variance, and

the method is computationally expensive; (iii) is a good compromise between the previous two [60]. Sec. 3.5 explains how our simulation combines these three methods with the hyperparameters grid search to provide an optimal and unbiased performance estimation and how we sample training- and validation-set to avoid OOS violation.

3.4.6. Validation metrics

As performance estimation metrics for binary classifiers, the literature presents a broad use of precision (1), recall (2), F1-score (3) and accuracy (4). Although these metrics are often sufficient, we introduce the measure of the area under the receiver operating characteristic curve (AUC). This curve describes the true-positive-rate (TPR) (2), which is another identification for the recall, as a function of the false-positive-rate (FPR) (5), within the domain of any possible FPR $\in [0, 1]$. We can derive these metrics directly from the confusion matrix, i.e., true positives (T_p), true negatives (T_n), false positives (F_p), and false negatives (F_n).

The binary BIBO classes are quite imbalanced and the classification task is rather challenging given the experiment's realistic conditions. We recreate a congested urban context with multiple buses operating at speed similar to walking pace, in proximity to various bus stops. Whereas F1-score identifies cases where the random classifier is better than our classifier, AUC identifies also cases where the classifier only predicts the larger class. The domain of precision (1), recall (2), F1-score (3), and accuracy (4) is $\in [0, 1]$, the higher the value the better. AUC's domain is also $\in [0, 1]$. The interpretation of AUC coefficient for random classifiers results in the same distribution of the F1-score, strictly around 0.5. In contrast with F1-score, for cases where classifiers predict only one class AUC presents the same distribution of the random classifier. Therefore, with AUC we expect good classifiers above 0.5 threshold, with higher values being better. Below this threshold a classifier would be consistent in predicting the wrong class. Both random and trivial classifiers should score 0.5 AUC in average. To assess our simulation results against both the random classifier and the single-class-predictor, AUC measures how well predictions are ranked, and is invariant to scale and classification-threshold [61]. Since at this stage we are agnostic on the cost of false positives and false negatives, these two properties are not a disadvantage, as opposed to the advantages in assessing the classification performance

with different levels of errors on the labels, over a large number of samples.

$$P = \frac{T_p}{T_p + F_p} \tag{1}$$

$$TPR = R = \frac{T_p}{T_p + F_n} \tag{2}$$

$$\text{F1} = 2\frac{P \cdot R}{P + R} \tag{3}$$

$$\text{A} = \frac{T_p + T_n}{Total\ population} \tag{4}$$

$$FPR = \frac{F_p}{T_n + F_p} \tag{5}$$

3.5. simulation of error distributions

After data preparation, as summarized in Alg. 1, we can proceed with the simulation (MS), as detailed in Alg. 3. In contrast to the literature studying flipping labels' problem using true ground-truth from a synthetic generation of datasets, we apply the following principles: We use high-quality ground-truth from a realistic setup involving real vehicles, devices, and people, producing real time-series from BLE and GPS sensors. We employ the method to simulate human errors as verified in the WoZ part of this experiment (see Sec. 3.2 and 3.3 for methodology and Sec. 4 for experimental evidence). We propagate such errors to the labels, assuming a state-of-the-art P2D ground-truth collection process, the same as real-world smartphone-based travel surveys.

Through the repeated sampling of user-by-user, the number of errors per user, and consequent propagation on the labels of each trip, at each repetition we train and evaluate ML classifiers against the random classifier, over features extracted from BLE sensors, versus features extracted from GPS sensors. We ensure the OOS validation principle on both grid-search search and methods' evaluation by randomly sampling 20% of the users and then picking all their trajectories to compose the validation set. Thus, we take the complement for the training-set.

We yield performance's unbiased estimation by applying a hold-out scheme within each run, where the training partition allows a grid-search through 5-fold cross-validation. Since the validation-set evaluation runs multiple times,

one for each draw, the results we obtain are comparable to a leave-one-out scheme or better, rather than the hold-out scheme. This process does not just flip labels, but also simulates the validation type of error in the experimental context (see Sec. 3.3). For example, if the user does not validate location and count of his or her alighting, we flip the BO labels of the corresponding trip-leg, to match the BI label of the previous trip-leg. However, in the simulations, we also apply random flip of labels from BI to BO and vice-versa, as possible sanity check algorithms on the labels are not in the scope if this work.

4. Results

In this sections we organize the results according to the research questions listed in Sec. 1.

4.1. People errors during ground-truth collection

The experiment included video recordings of the ground-truth for eighteen users in total, that we used for the P2D validation experiment. The resulting confusion matrix on error distributions for labels (see Table 3), shows that 50% of the received replies were perturbed. Nearly 60% of the user modified the counts, while the remaining population confirmed the counts as received. One user confirmed the perturbed count, and two users modified the correct counts. Overall, more than 40% of the validations contained at least one error, with average 0.7.

Table 3: Person to device ground-truth validation.

	Modified	Confirmed	Correct	Wrong	
Perturbed	6	1	3	4	7
Not Perturbed	2	5	5	2	7
	8	6	8	6	14
	14		14		

4.2. BIBO Classification Performance

Results show that the difference between the random classifier and the accelerometer-based activity recognition is minimal. Random forest trained and evaluated with camera-ground-truth performs significantly better when

classifying BLE beacons or GPS. In contrast, when processing features extracted from BLE, multi-layer perceptron performs worse than the random classifier in the same conditions. Overall, the low performance of production-level classifiers based on accelerometer reflects this challenging and realistic experiment setup. Although we only simulated an online classifier, and our output was off-line in practice, the BLE signal shows potential for the BIBO task. GPS yields the highest accuracy, as expected, but at the most expensive battery cost, compared to both accelerometer and BLE [8].

4.3. BIBO Resilience to Label Flipping

We repeated classifier training and evaluation with different error levels. Sampling from the same Poisson distribution we propagated errors to the ground-truth under two labeling-flip assumptions. In the first assumption, wrong users' counts will cause some segments to flip their correct class and match the previous or following segment's label. This assumption is consistent with the experiment setup. In the second assumption, more general, the discrete number of errors sampled from Poisson propagates to a random sample of trip segments by flipping the label from the correct class to the alternative. Under the first assumption, Figs. 3 and 5 show the AUC performance of each classifier using BLE beacons against the smartphones OS activity recognition, and the random classifier, for various errors levels. Similarly, Figs. 4 and 6 show the performance of each classifier using GPS sensors. Under the second assumption, Figs. 7 and 9 show the AUC performance using BLE; Figs. 8 and 10, using GPS.

When evaluating these classifiers on camera-ground-truth, after training on flipped labels at various rates, results suggest that RF are more sensitive to noisy labels when processing GPS features than when processing BLE features. The effect of noisy labels on multi-layer perceptron is negligible when processing GPS features; results show some slight performance improvements when errors propagate according to the first assumption.

We also note that in any case where high-quality ground-truth is not available, despite the "true" and "unknown" performance of these classifiers, the score is somewhat strongly biased, at a different rate according to the classifier. However, BLE signal combined with GPS and other sensors seem to have the potential of improving hybrid BIBO systems, more accurate and less energy-intensive.

Finally, we highlight that dealing with RSSI signal in the experimental context was challenging, and further work could enhance the process of

feature extraction from such a weak signal.

Table 4: Features [62] extracted from sensors' signals, within 10 seconds moving window: BLE RSSI and GPS Speed, Space- and Time-gap

1	Mean value
2	Max value
3	Min value
4	Position where the minimum value is located
5	Position where the maximum value is located
6	Amplitude between min and max value
7	Number of points beyond one standard deviation
8	Number of points below one standard deviation
9	Number of points above one standard deviation
10	Number of peaks in 10 seconds window
11	Number of peaks 5 seconds window
12	Number of peaks above 1 standard deviation
13	Peak distance within the same time window
14	Slope

5. Conclusion

This paper investigates the realistic large-scale deployment of a BIBO system based on BLE beacons, and analyzes the sensitivity of its ML components to errors on labels. The experimental setup recreates challenging conditions with a high density of BLE signals and low speeds of both users and vehicles present in the transport network.

We test our hypotheses on Poisson error distribution characterizing the labels collection process with person-to-device interactions, typical of current smartphone-based travel surveys. We find that users' validation errors affect both wrong and correct predictions. In the first case users are often unable to correct all the errors. In the second case, users introduce errors by amending correct predictions. Overall, users' did not improve significantly the ground-truth quality. Consequently, data cleansing process should take this factor into consideration beforehand.

We evaluated RF and MLP, first on BLE beacons signal and second on GPS. In addition, we evaluated the native Android and iOS classifiers, which rely mostly on the accelerometer. These classifiers comparison is based on

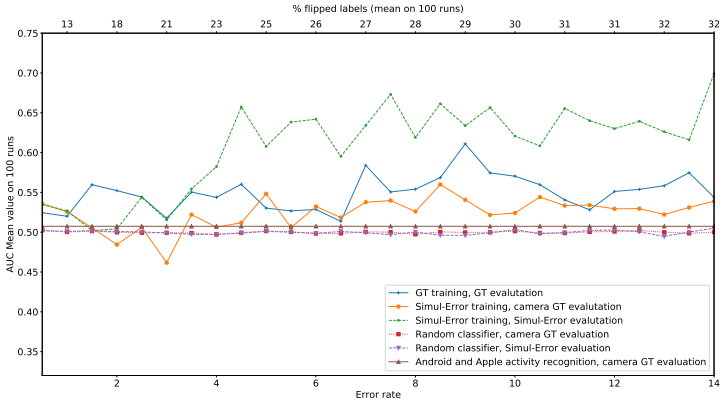

Figure 3: Random Forest one flip experiment
AUC Classification task using BLE RSSI signal only (p-values << 0.01)

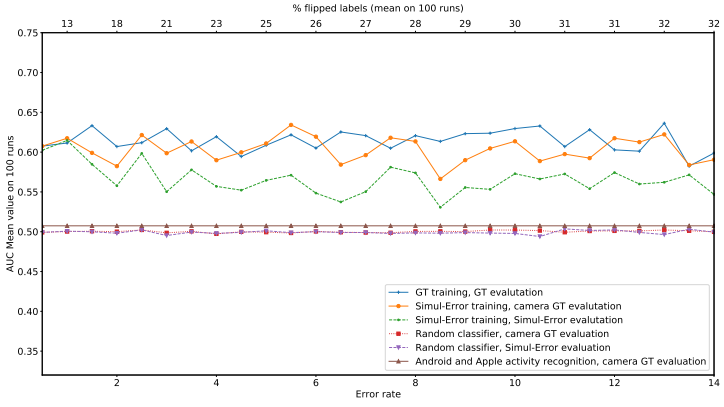

Figure 4: Random Forest one flip experiment
AUC Classification task using GPS signal only (p-values << 0.01)

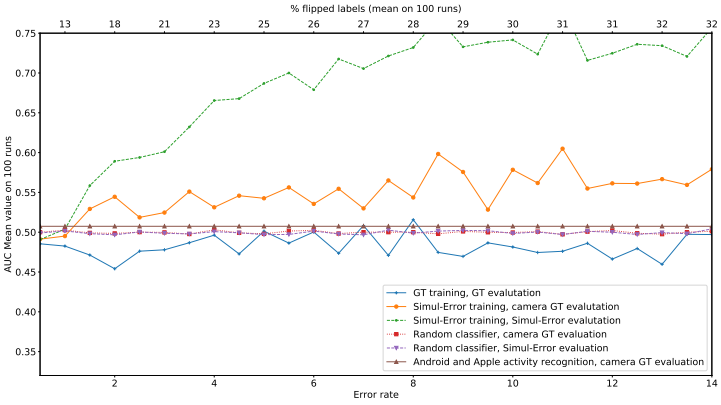

Figure 5: Multi Layer Perceptron one flip experiment
AUC Classification task using BLE RSSI signal only (p-values << 0.01)

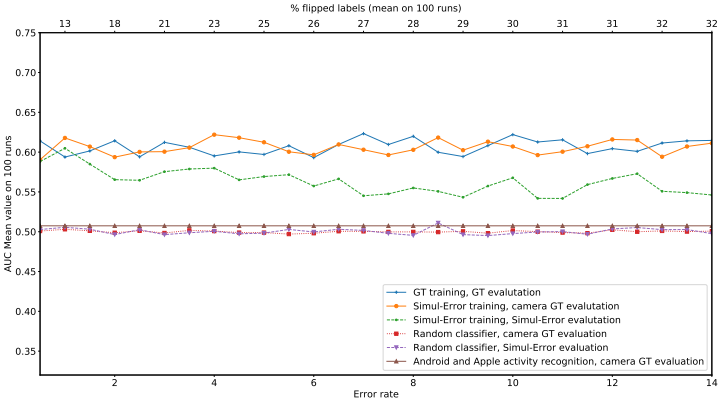

Figure 6: Multi Layer Perceptron one flip experiment
AUC Classification task using GPS signal only (p-values << 0.01)

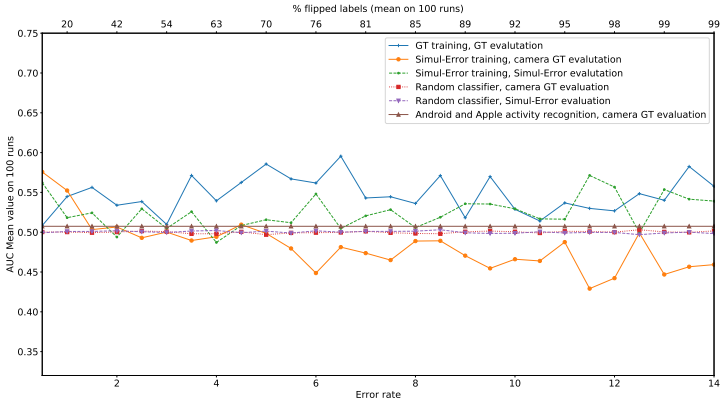

Figure 7: Random Forest two flip experiment
AUC Classification task using BLE RSSI signal only (p-values << 0.01)

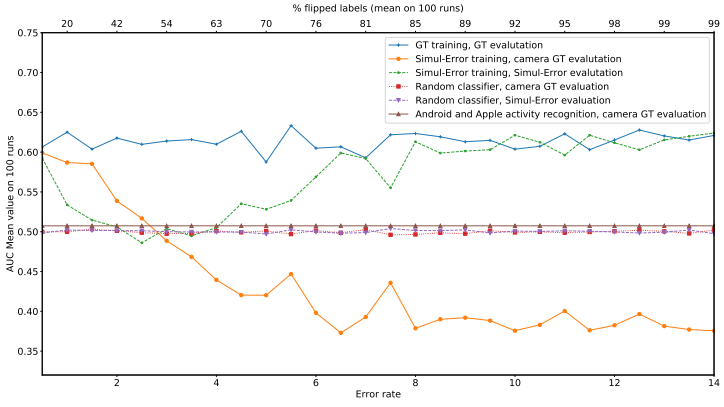

Figure 8: Random Forest two flip experiment
AUC Classification task using GPS signal only (p-values << 0.01)

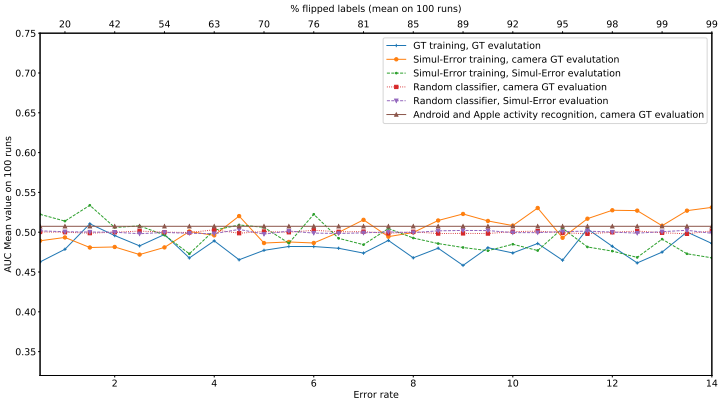

Figure 9: Multi Layer Perceptron two flip experiment
AUC Classification task using BLE RSSI signal only (p-values << 0.01)

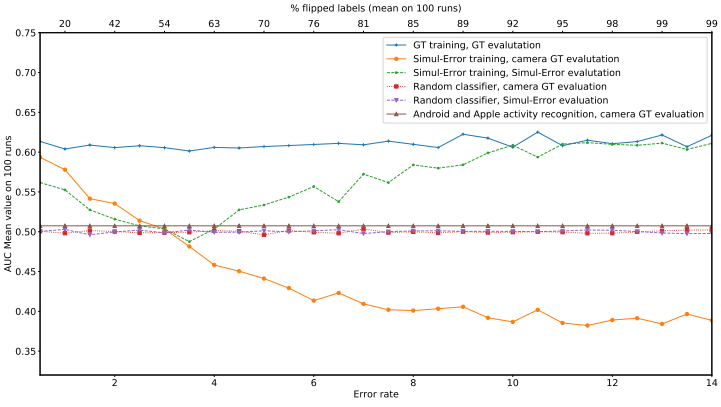

Figure 10: Multi Layer Perceptron two flip experiment
AUC Classification task using GPS signal only (p-values << 0.01)

AUC metric, which assigns the same score, 0.5, to both random classifiers and classifiers predicting one class only.

We find that off-the-shelf classifiers, based on the accelerometer, perform very close to the random classifier in this experimental context. Classifiers based on BLE beacons and GPS perform significantly better when trained with high-quality labels in the same context. When trained on noisy labels and evaluated on high-quality labels, at different levels, MLP seems more robust than RF on GPS features. Yet, when processing BLE features, MLP performance is below the random classifier. Overall, Random Forest performs significantly better on both BLE and GPS. At the same time, RF proves to be also robust to noise more on GPS than BLE.

When high-quality labels are unavailable—even when the noise rate is relatively low—and classifiers are trained and evaluated blindly, the classifiers' evaluation yields a significant and large bias level, underestimating or overestimating the real performance. This problem may affect many results in the available literature. Therefore, efforts will be directed in two directions. One for supervised classifiers such as RF, towards the development of methodologies to assess performance sensitivity on labels' quality. Another for MLP and neural networks, towards architectures able to reduce or eliminate the dependency from labels.

6. Imputation

Fig. 11 shows that compared to the total points collected by the sensing platform, BLE beacons readings are present only on a fraction of the points where GPS is present; the rest of the points are empty. BLE signal goes undetected when the receiver device is not in the beacon range. However, the relative position of the two devices to the user's body often leads to the same result even when the two are in range [11]. Therefore, we need to perform imputation and fill the gaps whenever appropriate. Existing work shows multiple techniques. Although Kalman-filters might seem the obvious choice from indoor experience [32], this use case requires a faster and relatively more trivial method. From this standpoint, we consider exponential-weighted-moving-average (EWMA), which consists of computing the average of the readings within a time window, where points close to the center window have a higher weight than points at the end of the window [63]. For EWMA, the weight depends on the window size and the decay rate. From the perspective of a fingerprinting approach (see Sec. 2.1), especially on large-scale

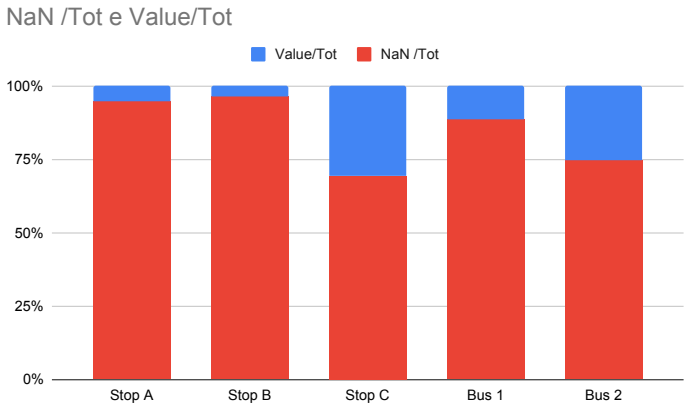

Figure 11: Beacons RSSI timestamp with values vs. Not a Number (NaN), on total points available.

deployments, we need to inform the classifier on points where the imputation algorithm could not fill the gaps. We cannot use zero, because BLE beacons signal domain can be found, empirically, in the following domain $RSSI \in (-100, -50)$ [64]. Further, smartphones record the null value when on the fringe of a BLE range, which is counter-intuitive given the signal's domain. Therefore, because of the meaning of null value and the expected amount of gaps, filling these gaps with zero is likely to poison any classifier. Instead, we can fill these positions with an arbitrary constant and augment the fingerprint vector reporting a weight 0 in the position filled with the arbitrary constant and 1 otherwise [65]. We call this step imputation trick. (6) defines the fingerprint vector at time t as FP_t, where v_i represents the RSSI signal received from the i^{th} BLE beacon, while $v_{j_{GAP}}$ is the gap of signal from the j^{th} BLE beacon. To account for gaps during the learning process, and avoid poisoning the classifier, $FP_t \in R^{m+n}$ can be augmented, resulting in a new vector $FPA_t \in R^{2\cdot(m+n)}$ (7), where $v_{j_{IMP}}$ correspond to the signal imputation of the j^{th} BLE beacon gap, for example with EWMA, while $v_{k_{CONST}}$ represent the remaining signal gap from the j^{th} BLE beacon, filled with an arbitrary not null constant. We pass this information to the classifier through the aforementioned binary part of the augmented vector FPA_t.

$$
\begin{aligned}
FP_t = &(v_0, v_1, \ldots, v_{0_{GAP}}, \\
&\ldots, v_i, \ldots, v_{n_{GAP}}, \\
&\ldots, v_m), \\
&m > 0 \wedge n \geq 0 \wedge i \in (1, m)
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
FPA_t = &(v_0, v_1, \ldots, \\
&v_{0_{IMP}}, \ldots, v_i, \ldots, v_{n_{IMP}}, \ldots, \\
&v_{0_{CONST}}, \ldots, v_j, \ldots, v_{k_{CONST}}, \\
&\ldots, v_m, \\
&1_0, 1_1, \ldots \\
&1_0, \ldots, 1_i, \ldots, 1_n, \ldots, \\
&0_0, \ldots, 1_j, \ldots, 0_k \\
&\ldots, 1_m), \\
&m > 0 \wedge n \geq 0 \wedge k \geq 0 \\
&\wedge i, j \in (1, m), i \neq j \\
&v_{s_{CONST}} = v_{p_{CONST}} = C, \\
&\forall s, p \in [0, k], s \neq p
\end{aligned}
\tag{7}
$$

7. Algorithms

This section lists the pseudo-code of the algorithms implemented for the simulation. Alg. 1 refers to the data preparation and Alg. 2 to the error simulation and propagation. Alg. 3 encompasses both Alg. 1 and 2, and performs the following steps.

1. Iterative grid-search of the optimal hyperparameteres, accomplished only once per setting, at loop $C = 0$.

2. Model Training, accomplished at each loop $C \geq 0$, using the same optimal hyperparameters found at loop $C = 0$.

3. Model Evaluation, accomplished the four settings of interest.

These four settings of interest are the following.

1. evaluation on camera GT of the model trained with camera GT;
2. evaluation on GT with flipped labels of the model trained on GT with flipped labels;
3. evaluation on camera GT of the model trained on GT with flipped labels.
4. evaluation of a random classifier on camera GT.

Algorithm 1: Data preparation

Result: Clean trajectories, assign trip IDs, and extract standardized features for both GPS and BLE signals

Input : raw dataset (RD), true labels (TL)

Output: dataset with tripID labels and features vectors (CD)

$UULIST \leftarrow$ list-unique-users(RD)
foreach $user \in UULIST$ **do**
  $tripIDs_{\text{user}} \leftarrow$ clean-segment-trajectories(RD, user, TL)
  foreach $TS \in \{BLE, GPS\}$ **do**
    if $TS == BLE$ **then**
      $CD_{user} \leftarrow$ imputation-trick(D, user, TS, $tripIDs_{\text{user}}$)
    end
    $CD_{user} \leftarrow$ extract-standard-features(CD_{user}, TS)
  end
  CD.insert(CD_{user})
end
return CD

Algorithm 2: Simulate and propagate P2D validation errors

Result: Faulty ground-truth vector

Input : true labels vector (TL), unique users list (UULIST), features-from-pre-processed-dataset (FCD, see Alg. 1)

Output: flipped labels vector FL

foreach *user* $\in$ *UULIST* **do**
    `/* draw errors number from Poisson distribution */`
    $NE \leftarrow$ draw-from-Poisson(ERR)
    `/* Draw NE random TripIDs, as mislabeled trips */`
    WrongTID$_{user} \leftarrow$ draw-random-tripIDs(NE, FCD_{user})
    `/* Copy TL and flip labels for each trip drawn in the previous step */`
    $FL \leftarrow$ TL
    foreach *trip* $\in$ *WrongTID*$_{user}$ **do**
        $FL_{trip} \leftarrow$ flip-labels(FL_{trip})
    end
end
return FL

Algorithm 3: Model/Sensor performance estimation

Result: BIBO Performance distributions of RF and MLP models, evaluated separately for BLE and GPS features, over different average error rates

Input : features-from-pre-processed-dataset (FCD, see Alg. 1), true-labels (TL), target-signal (TS), hyperparameters-search-space (HSS, see Table 1, 2), maximum-error-rate (MERR)

Output: F1 (3), A (4), AUC, Optimal Hyperparameters (OP)

```
/* Simulate flipping labels and evaluate model performance against true ground-truth                    */
UULIST ← list-unique-users(FCD)
ERR ← 0.5
while ERR ≤ MERR do
    while C < 100 do
        /* Simlulate users errors and propagate through trajectory labels (see Alg. 2)                  */
        FL ← simulate-and-propagate-error(UULIST, TL, FCD)
        /* Create Training- and Validation-set, compliant with OOS principle                            */
        VA ← pick-random-user-IDs(UULIST, users-num=2)
        VA ← extract-features-trajectories-by-user-ID-from-dataset(FCD,VA)
        TR ← extract-features-trajectories-by-user-ID-from-dataset(FCD,VA^C)
        /* Evaluate classifiers against true and flipped labels (TL Vs. FL)                             */
        foreach (TR,VA) ∈ {(TR,VA)_GPS, {(TR,VA)_BLE} do
            foreach model ∈ {RF, MLP} do
                foreach label ∈ {FL, TL} do
                    L ← label
                    M ← model
                    if C=0 then
                        /* Hyperparameters 5-fold grid-search on training-set                           */
                        OP_L, (F1_{L_CV}, A_{L_CV}, AUC_{L_CV})_M ← model-5-fold-grid-search(TR,L)
                    else
                        /* Train a classifier with optimal hyperparameters and labels L                 */
                        classifier_L ← train-model(TR,L,OP_L)
                    end
                    /* Hold-out evaluation on true labels and validation-set, of a classifier M trained
                       with input-labels L                                                              */
                    (F1_{L_HO}, A_{L_HO}, AUC_{L_HO})_M ← evaluate-model(classifier_L, VA, TL)
                    /* Hold-out evaluation on flipped labels and validation-set, of a classifier M
                       trained with input-labels L                                                      */
                    (F1_{L_HO}, A_{L_HO}, AUC_{L_HO})_M ← evaluate-model(classifier_L, VA, FL)
                    /* Hold-out evaluation on labels L and validation-set, of random classifier         */
                    (F1_{RL_HO}, A_{RL_HO}, AUC_{RL_HO})_M ← evaluate-model(random, VA, L)
                    (F1, A, AUC, OP).insert(F1, A, AUC, OP)_M
                end
            end
        end
        C ← C+1
    end
    ERR ← ERR+0.5
end
return ( OP , F1 , A , AUC )
```

Acknowledgment

This project is co-financed by the European Regional Development Fund through the Urban Innovative Actions Initiative.

References

[1] B. D. Hankin, R. A. Wright, Passenger flow in subways, Journal of the Operational Research Society 9 (1958) 81–88. doi:10.1057/jors.1958.9.

[2] J. Zhang, D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, Z. Li, A real-time passenger flow estimation and prediction method for urban bus transit systems, Ieee Transactions on Intelligent Transportation Systems 18 (2017) 7898469. doi:10.1109/TITS.2017.2686877.

[3] F. Y. Wang, Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications, Ieee Transactions on Intelligent Transportation Systems 11 (2010) 5549912. doi:10.1109/TITS.2010.2060218.

[4] J. Dekkers, P. Rietveld, Electronic ticketing in public transport: A field study in a rural area, Journal of Intelligent Transportation Systems 11 (2007) 69–78. URL: https://doi.org/10.1080/15472450701293866. doi:10.1080/15472450701293866. arXiv:https://doi.org/10.1080/15472450701293866.

[5] M. Mezghani, Study on electronic ticketing in public transport, 2008. URL: https://emta.com/IMG/pdf/EMTA-Ticketing.pdf.

[6] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselbock, N. Hofler, Be-in/be-out with bluetooth low energy: Implicit ticketing for public transportation systems, Ieee Conference on Intelligent Transportation Systems, Proceedings, Itsc 2015- (2015) 7313345. doi:10.1109/ITSC.2015.253.

[7] Global mobile market report (2020).

[8] V. Servizi, F. C. Pereira, M. K. Anderson, O. A. Nielsen, Transport behavior-mining from smartphones: a review, European Transport Research Review 13 (2021) 57. URL: https://doi.org/10.1186/s12544-021-00516-z. doi:10.1186/s12544-021-00516-z.

[9] Y. Cui, S. S. Ge, Autonomous vehicle positioning with gps in urban canyon environments, Ieee Transactions on Robotics and Automation 19 (2003) 15–25. doi:10.1109/TRA.2002.807557.

[10] P. Sapiezynski, A. Stopczynski, D. K. Wind, J. Leskovec, S. Lehmann, Inferring person-to-person proximity using wifi signals, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1 (2017). URL: https://doi.org/10.1145/3090089. doi:10.1145/3090089.

[11] C. Li, P. C. Zegras, F. Zhao, Z. Qin, A. Shahid, M. Ben-Akiva, F. C. Pereira, J. Zhao, Enabling bus transit service quality co-monitoring through smartphone-based platform, Transportation Research Record 2649 (2017) 42–51. doi:10.3141/2649-05.

[12] D. Rolnick, A. Veit, S. Belongie, N. Shavit, Deep learning is robust to massive label noise (2018).

[13] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, R. Fergus, Training convolutional networks with noisy labels, 2015. arXiv:1406.2080.

[14] A. Ahmed, H. Yousif, R. Kays, Z. He, Animal species classification using deep neural networks with noise labels, Ecological Informatics 57 (2020) 101063. doi:10.1016/j.ecoinf.2020.101063.

[15] D. Hendrycks, M. Mazeika, D. Wilson, K. Gimpel, Using trusted data to train deep networks on labels corrupted by severe noise, 2018.

[16] N. Natarajan, I. S. Dhillon, P. Ravikumar, A. Tewari, Learning with noisy labels (2016). doi:10.1.1.884.8471.

[17] K. Yi, J. Wu, Probabilistic end-to-end noise correction for learning with noisy labels, Proceedings of the Ieee Computer Society Conference on Computer Vision and Pattern Recognition 2019- (2019) 8953202. doi:10.1109/CVPR.2019.00718.

[18] D. F. Nettleton, A. Orriols-Puig, A. Fornells, A study of the effect of different types of noise on the precision of supervised learning techniques, Artificial Intelligence Review 33 (2010) 275–306. doi:10.1007/s10462-010-9156-z.

[19] F. Rodrigues, F. Pereira, Deep learning from crowds (2017) 10.

[20] Y. Liu, H. Guo, Peer loss functions: Learning from noisy labels without knowing noise rates, 2019.

[21] L. Jiang, D. Huang, M. Liu, W. Yang, Beyond synthetic noise: Deep learning on controlled noisy labels (2020).

[22] N. Brouwers, M. Woehrle, Dwelling in the canyons: Dwelling detection in urban environments using gps, wi-fi, and geolocation, Pervasive and Mobile Computing 9 (2013) 665–680. doi:10.1016/j.pmcj.2012.07.001.

[23] K. Muthukrishnan, B. J. Van Der Zwaag, P. Havinga, Inferring motion and location using wlan rssi, Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5801 (2009) 163–182. doi:10.1007/978-3-642-04385-7_12.

[24] M. Y. Mun, D. Estrin, J. Burke, M. Hansen, Parsimonious mobility classification using gsm and wifi traces. hot-emnets (2011). doi:10.1.1.183.5677.

[25] P. A. Gonzalez, J. S. Weinstein, S. J. Barbeau, M. A. Labrador, P. L. Winters, N. L. Georggi, R. Perez, Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks, Iet Intelligent Transport Systems 4 (2010) 37–49. doi:10.1049/iet-its.2009.0029.

[26] D. K. Wind, P. Sapiezynski, M. A. Furman, S. Lehmann, Inferring stop-locations from wifi, Plos One 11 (2016) e0149105. doi:10.1371/journal.pone.0149105.

[27] A. Bjerre-Nielsen, K. Minor, P. Sapiezynski, S. Lehmann, D. D. Lassen, Inferring transportation mode from smartphone sensors: Evaluating the potential of wi-fi and bluetooth, Plos One

15 (2020). doi:10.1371/journal.pone.023400310.1371/journal.pone.0234003.r00110.1371/journal.pone.0234003.r00210.1371/journal.pone.0234003.r00310.1371/journal.pone.0234003.r00410.1371/journal.pone.0234003.r005.

[28] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, G. K. Karagiannidis, A survey on mobile anchor node assisted localization in wireless sensor networks, Ieee Communications Surveys and Tutorials 18 (2016) 7438736. doi:10.1109/COMST.2016.2544751.

[29] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, E. Aboutanios, Recent advances in indoor localization: A survey on theoretical approaches and applications, Ieee Communications Surveys and Tutorials 19 (2017) 7762095. doi:10.1109/COMST.2016.2632427.

[30] A. Kotanen, M. Hännikäinen, H. Leppäkoski, T. D. Hämäläinen, Experiments on local positioning with bluetooth, Proceedings Itcc 2003, International Conference on Information Technology: Computers and Communications (2003) 1197544. doi:10.1109/ITCC.2003.1197544.

[31] F. Subhan, H. Hasbullah, A. Rozyyev, S. T. Bakhsh, Indoor positioning in bluetooth networks using fingerprinting and lateration approach, 2011 International Conference on Information Science and Applications, Icisa 2011 (2011) 5772436. doi:10.1109/ICISA.2011.5772436.

[32] L. Chen, H. Kuusniemi, Y. Chen, J. Liu, L. Pei, L. Ruotsalainen, R. Chen, Constraint kalman filter for indoor bluetooth localization, 2015 23rd European Signal Processing Conference, Eusipco 2015 (2015) 7362717. doi:10.1109/EUSIPCO.2015.7362717.

[33] F. Subhan, H. Hasbullah, K. Ashraf, Kalman filter-based hybrid indoor position estimation technique in bluetooth networks, International Journal of Navigation and Observation 2013 (2013) 570964. doi:10.1155/2013/570964.

[34] H. J. Pérez Iglesias, V. Barral, C. J. Escudero, Indoor person localization system through rssi bluetooth fingerprinting, 2012 19th International Conference on Systems, Signals and Image Processing, Iwssip 2012 (2012) 6208163.

[35] A. Moubayed, A. Shami, Softwarization, virtualization, machine learning for intelligent effective v2x communications, Ieee Intelligent Transportation Systems Magazine (2020) 14. doi:10.1109/MITS.2020.3014124.

[36] Apple BLE Detection API, 2021. URL: https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/LocationAwarenessPG/RegionMonitoring/RegionMonitoring.

[37] E. Beigman, B. B. Klebanov, Learning with annotation noise, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1, ACL '09, Association for Computational Linguistics, USA, 2009, p. 280–287.

[38] N. Manwani, P. S. Sastry, Noise tolerance under risk minimization, IEEE Transactions on Cybernetics 43 (2013) 1146–1151. doi:10.1109/TSMCB.2012.2223460.

[39] C. Man Teng, A comparison of noise handling techniques (2015). doi:10.1.1.529.9973.

[40] R. Barandela, E. Gasca, Decontamination of training samples for supervised pattern recognition methods, in: F. J. Ferri, J. M. Iñesta, A. Amin, P. Pudil (Eds.), Advances in Pattern Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 621–630.

[41] C. E. Brodley, M. A. Friedl, Identifying mislabeled training data, Journal of Artificial Intelligence Research 11 (1999) 131–167. doi:10.1613/jair.606.

[42] A. L. Miranda, L. P. F. Garcia, A. C. Carvalho, A. C. Lorena, Use of classification algorithms in noise detection and elimination, Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5572 (2009) 417–424. doi:10.1007/978-3-642-02319-4_50.

[43] B. Yuan, J. Chen, W. Zhang, H. Tai, S. McMains, Iterative cross learning on noisy labels, in: 2018 IEEE Winter Conference on Applications

of Computer Vision (WACV), 2018, pp. 757–765. doi:10.1109/WACV.2018.00088.

[44] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, R. Fergus, Training convolutional networks with noisy labels, 3rd International Conference on Learning Representations, Iclr 2015 - Workshop Track Proceedings (2015).

[45] I. Jindal, M. Nokleby, X. Chen, Learning deep networks from noisy labels with dropout regularization, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), 2016, pp. 967–972. doi:10.1109/ICDM.2016.0121.

[46] T. Xiao, T. Xia, Y. Yang, C. Huang, X. Wang, Learning from massive noisy labeled data for image classification, Proceedings of the Ieee Computer Society Conference on Computer Vision and Pattern Recognition 07-12- (2015) 7298885. doi:10.1109/CVPR.2015.7298885.

[47] D. P. Kingma, D. J. Rezende, S. Mohamed, M. Welling, Semi-supervised learning with deep generative models, Advances in Neural Information Processing Systems 4 (2014) 3581–3589.

[48] X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation (2009). doi:10.1.1.13.8280.

[49] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning object categories from internet image searches, Proceedings of the IEEE 98 (2010) 1453–1466. doi:10.1109/JPROC.2010.2048990.

[50] Apple Mode Detection API, 2021. URL: https://developer.apple.com/documentation/coremotion/cmmotionactivity.

[51] Android Mode Detection API, 2021. URL: https://developers.google.com/location-context/activity-recognition.

[52] E. Cascetta, S. Nguyen, A unified framework for estimating or updating origin/destination matrices from traffic counts, Transportation Research, Part B (methodological) 22B (1988) 437–55. doi:10.1016/0191-2615(88)90024-0.

[53] L. D. Riek, Wizard of oz studies in hri, Journal of Human-robot Interaction 1 (2012) 119–136. doi:10.5898/JHRI.1.1.Riek.

[54] V. Servizi, N. C. Petersen, F. C. Pereira, O. A. Nielsen, Stop detection for smartphone-based travel surveys using geo-spatial context and artificial neural networks, Transportation Research Part C: Emerging Technologies 121 (2020) 102834. doi:10.1016/j.trc.2020.102834.

[55] S. Dabiri, K. Heaslip, Inferring transportation modes from GPS trajectories using a convolutional neural network, Transportation Research Part C: Emerging Technologies 86 (2018) 360–371. URL: https://doi.org/10.1016/j.trc.2017.11.021. doi:10.1016/j.trc.2017.11.021.

[56] X. Zhou, P. L. K. Ding, B. Li, Improving robustness of random forest under label noise, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 950–958. doi:10.1109/WACV.2019.00106.

[57] L. Breiman, Manual on setting up, using, and understanding random forests v4.1, 2002.

[58] C. M. Bishop, Pattern Recoginiton and Machine Learning, 2006.

[59] H. Tim, New perspectives on the performance of machine learning classifiers for mode choice prediction, Report TRANSP-OR (2020). URL: https://transp-or.epfl.ch/documents/technicalReports/HillelNew2020.pdf.

[60] A. Baraldi, L. Bruzzone, P. Blonda, Quality assessment of classification and cluster maps without ground truth knowledge, Ieee Transactions on Geoscience and Remote Sensing 43 (2005) 857–872. doi:10.1109/TGRS.2004.843074.

[61] G. Cantarero, R. Jarabo, The area under the roc curve, Medicina Clinica 106 (1996) 355–356.

[62] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, N. S. Jones, catch22: Canonical time-series characteristics, Data Mining and Knowledge Discovery 33 (2019) 1821–1852.

[63] L. Shu, Y. Su, W. Jiang, K.-L. Tsui, A comparison of exponentially weighted moving average-based methods for monitoring increases in incidence rate with varying population size, IIE Transactions 46 (2014) 798–812. URL: https://doi.org/10.1080/0740817X.2014.894805. doi:10.1080/0740817X.2014.894805. arXiv:https://doi.org/10.1080/0740817X.2014.894805.

[64] J. Paek, J. Ko, H. Shin, A Measurement Study of BLE iBeacon and Geometric Adjustment Scheme for Indoor Location-Based Mobile Applications, Mobile Information Systems 2016 (2016) 8367638. URL: https://doi.org/10.1155/2016/8367638. doi:10.1155/2016/8367638.

[65] I. Malmberg, An analysis of iBeacons and critical minimum distances in device placement, Ph.D. thesis, 2014. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-187925.

5 Paper D: Large Scale Passenger Detection with Smartphone/Bus Implicit Interaction and Multisensory Unsupervised Cause-effect Learning

The following pages contain the article:

V. Servizi, D. R. Persson, F. C. Pereira, H. Villadsen, P. Bækgaard, J. Rich, and O. A. Nielsen (2021). "Large Scale Passenger Detection with Smartphone/Bus implicit interaction and Multisensory Unsupervised Cause-effect learning". In: *IEEE Transactions on Intelligent Transportation Systems (UNDER REVIEW)*.

Please cite accordingly.

The work was part of poster presentations at the *"Grand Opening of DTU Centre for Collaborative Autonomous Systems and Autonomous Systems Test Arena"*, Copenhagen, October, 2020.

The work was also part of a presentation at the *"Annual Transport Conference at Aalborg University"*, Aalborg, August, 2021.

Large Scale Passenger Detection with Smartphone/Bus Implicit Interaction and Multisensory Unsupervised Cause-effect Learning

Valentino Servizi[a,*], Dan R. Persson[b], Francisco C. Pereira[a], Hannah Villadsen[c], Per Bækgaard[b], Jeppe Rich[a], Otto A. Nielsen[a]

[a]*Department of Technology, Management and Economics*
Technical University of Denmark (DTU)
Kgs. Lyngby Denmark
[b]*Department of Applied Mathematics and Computer Science*
DTU
[c]*Department of People and Technology*
Roskilde University

Abstract

Intelligent Transportation Systems (ITS) underpin the concept of Mobility as a Service (MaaS), which requires universal and seamless users' access across multiple public and private transportation systems while allowing operators' proportional revenue sharing. Current user sensing technologies such as Walk-in/Walk-out (WIWO) and Check-in/Check-out (CICO) have limited scalability for large-scale deployments. These limitations prevent ITS from supporting analysis, optimization, calculation of revenue sharing, and control of MaaS comfort, safety, and efficiency. We focus on the concept of implicit Be-in/Be-out (BIBO) smartphone-sensing and classification.

To close the gap and enhance smartphones towards MaaS, we developed a proprietary smartphone-sensing platform collecting contemporary Bluetooth Low Energy (BLE) signals from BLE devices installed on buses and Global Positioning System (GPS) locations of both buses and smartphones. To enable the training of a model based on GPS features against the BLE pseudo-label, we propose the Cause-Effect Multitask Wasserstein Autoencoder (CEMWA). CEMWA combines and extends several frameworks around Wasserstein autoencoders and neural networks. As a dimensionality reduction tool, CEMWA obtains an auto-validated representation of a latent space describing users' smartphones within the transport system. This representa-

tion allows BIBO clustering via DBSCAN.

We perform an ablation study of CEMWA's alternative architectures and benchmark against the best available supervised methods. We analyze performance's sensitivity to label quality. Under the naïve assumption of accurate ground truth, XGBoost outperforms CEMWA. Although XGBoost and Random Forest prove to be tolerant to label noise, CEMWA is agnostic to label noise by design and provides the best performance with an 88% F1 score.

Keywords: Device-to-device, Sensor-to-sensor, Ground-truth-validation, Wasserstein-auto-encoders, Autonomous-vehicles

1. Introduction

Tracking passenger movements through the public transport network, seamlessly and without direct human interaction, requires accurate models and methods to discriminate between passengers that are using the public transport network and anyone else outside the transport network. While the accurate solution of such an implicit Be-In/Be-Out (BIBO) classification problem [1], is directly relevant as a mean to collect important data from the public transport system, e.g. Check-in/Check-out or Walk-in/Walk-out statistics, it is relevant for other areas as well. This includes as an example, the tracking of persons entering buildings to comply with safety measures and the registration, and tracking of people in supermarkets to support crew management in different parts of the supermarket. However, tracking of public transport users represent a more complex problem in that buses and passengers move in space- and time. As a result, we will argue that the ability to provide robust solutions for public transport applications is a stepping-stone for these other relevant applications.

Solving the before mentioned classification problem is important for several reasons. Firstly, on the very practical side it provides a means to collect valuable data about passenger flows that would otherwise have been lost for users paying by cash, or accidentally traveling without checking-in. Secondly, it would enable context-aware surveying and services while lifting the burden

*Corresponding author. Email: valse@dtu.dk

of explicit interaction from passengers. Thirdly, for planning optimal departure times and routes of a trip through the public network, it would support personalized dynamic recommendations.

In a wider perspective the presented methodology can be seen as an important component in Mobility-as-a-service (MaaS) systems. MaaS combines multiple transport modes as transport services–e.g., car, bus, bike, scooter– offered through a single interface, and paid with the same unique subscription, as the media contents on "Netflix" [2, 3]. Hence, MaaS is essentially *"a data-driven, user-centered paradigm, powered by the growth of smartphones"* [4]. Regardless from the perspective, MaaS ultimate goal is to enable a door-to-door public service, attractive for the passengers, and competitive with, e.g., privately owned cars. In this context, the ability to accurately track passengers while traveling would underpin the efficient capacity planning for a dynamic, responsive, and intelligent public transport paradigm.

In the MaaS context, smartphone-based automatic fare collection systems (AFCS) with BIBO could allow the integration of public service ticketing, automatic price calculation, and a fair cost split across multiple operators. The latter point includes emerging providers of, e.g., car- and bike-sharing services. Compared to CICO and WIWO, BIBO offers at least two advantages: (i) public transit increased comfort for passengers [5]; and (ii) operational integration mostly software, with a negligible impact on new physical infrastructure. The second point means potentially lower access barriers for emerging transport service providers to MaaS. For the first, we refer to the passengers increased comfort with the term ticketless. Ticketless identifies the perspective of a system ability to flexibly adapting the transport service bill to the user's journey(s) across multiple service providers, as opposed to the perspective of multiple tickets necessary from multiple service providers, for the same journey.

From the Big Data perspective, handling this binary classification problem with supervised machine learning methods presents the following challenges:

1. Controlling noise in the labels:
2. Operating a sustainable labels collection cost;
3. Minimizing the impact of sensors and data collection on the battery; and

4. Minimizing the users' privacy exposure.

These challenges involve the service operator's perspective in the first case and the smartphone user's perspective in the others.

Although from a ticketing perspective there should be no noise, thus one should only be charged when he or she uses a transport service, when using tickets as labels to train machine learning algorithms, the assumption of possible undetected ticketing errors from both sides–passenger and service provider–seems more than reasonable.

Mining transport behavior from smartphones data relies, among other sensors, on Global Positioning System (GPS), Inertial Navigation System (INS), and Bluetooth Low Energy (BLE) signal[6]. In urban areas, where 80% of public transport demand occurs [7] (e.g., in Denmark), the classification of sensors' observations is complex. With GPS, any transportation mode looks the same due to a combination of factors, such as GPS errors in urban canyons, proximity between pedestrians and buses, and vehicles' low speeds in congested traffic [8]. With INS, multiple habits, each corresponding to whether one carries a smartphone, e.g., in the pocket or the bag, determine different sensors patterns [9]; the integral of any noise included in the sensors' signal, in addition, leads to often unmanageable error drifts [10]. The BLE signal, which is extensively studied for indoor tracking, presents an excellent potential for proximity sensing and battery efficiency [11]. However, smartphones' signal records of BLE devices in proximity suffer from signal gaps [12]; a higher spatial density of BLE devices allows good indoor-tracking performance, but such a density is not scalable at a city scale. In contrast, GPS and INS scaling potential correspond to a heavy impact on the smartphones' battery [6]. In the first case, the sensor is directly responsible for the energetic consumption. In the second case, the sensors' energy consumption is sustainable as long as the signals are classified online within the smartphone. Yet, due to the high sampling rate necessary for achieving acceptable classification performance, $> 20\,\mathrm{Hz}$, data consumption outside the smartphone would imply high network energy consumption for data transfer [6]. In the assumption of training a supervised machine learning algorithm with high-quality labels, BIBO binary classification in the urban context seems a difficult task. When labels' quality degrades, we face another limitation as classifiers' performance can be highly biased—consequently, decisions would be based on scores looking high when they are low in reality and vice-versa [13]. To overcome the limitations mentioned above, in this work, we rely on

a unique dataset collected during three months of autonomous buses' operations across a local public network in Denmark. The dataset includes the GPS and BLE trajectories collected from buses and passengers' smartphones through a proprietary smartphone-sensing platform, including 300 BLE devices installed in buildings near the bus network, in the buses, and at bus stops. Another set of the data provides high-quality ground truth collected by users that followed precise instructions on individual sequences of origins and destinations within the bus network, along specific routes [14].

1.1. Literature Review

The solution we propose for the BIBO classification problem involves the implicit interaction of passenger smartphones, buses, and bus-network [13, 1]. Therefore, it falls within the intersection of several disciplines converging around smartphone-based travel surveys and smartphones indoor tracking with BLE network interaction. In the first case, leveraging smartphone onboard sensors, we are interested in the limitations of the methods for mode detection in general and bus detection in particular [5]; in the second case, we are interested in how to deal with BLE signals [13].

The literature on mode detection from smartphones data is pervasive. GPS and INS sensors are the most used also to provide location- and person-agnostic mode classification. GPS and INS systems generate very different trajectories. The first system provides a geospatial time series with a sampling rate $\geq 1\,\mathrm{Hz}$ [15, 16]; the second system, a three-dimension time series along the three axes of the smartphone's reference frame, and a sampling rate $\geq 20\,\mathrm{Hz}$ [6, 17]. To prepare the data for the classification, the steps one follows to clean and segment these trajectories differ too. However, the best-performing classification methods consist of two main groups. The first group includes supervised methods, such as decision trees, random forest, and XGBoost [18]; the second group has various configurations of artificial neural networks (ANN), both supervised and semi-supervised. Unsupervised methods based on clustering are applied directly to features extracted from GPS and INS, but their performance seems below the supervised and semi-supervised methods mentioned above. The blooming literature on both GPS- and INS-based mode detection proposes very effective methodologies, equally accurate when datasets include urban and outskirt areas and multiple transportation targets [6]. However, at low speeds, state-of-the-art INS-based online classifiers available on the leading smartphone operation systems seem

unable to discriminate between bus and walk mode. In contrast, GPS and BLE classifiers show higher performance [13].

Among the studies focusing on mode detection and public transportation, specifically buses, the most promising are considering the interaction between users and the transport network. This interaction could be expressed as the time series of the distances between each point of a smartphone's GPS trajectory and each point of interest (PoI) extracted from the infrastructure mapped on GIS [19]. The classification could be point-based, thus relying on short segments. Another approach, which we define segment-based [6], could look at longer trip segments and the periodicity of stops typical of any bus operation [20]. However, while the first approach suffers the limitation from the GPS error in dense urban areas, the second approach seems ineffective for short trips.

Literature focusing on BLE and WiFi signals–both based on the same communicaiton frequency and protocols sharing some similarities–converges between indoor tracking and mode detection. The traditional methodologies leverage the Friis equation, and the trilateration [21, 22]. However, machine learning methods such as random forests and Gaussian processes are effective in BLE or WiFi fingerprint classification, and spatial signal mapping [23, 24, 25]. To allow optimal BIBO sensing and classification with BLE devices, we find no clear contributions on the minimum spatial density of BLE devices, nor how to cover the scale of a city [13]. Therefore, we rely on literature about indoor tracking [26] and preliminary BIBO experiments with BLE signals [13], suggesting that BLE devices installed in buses and bus stops could offer a coverage sufficient for classification. Consequently, such a configuration would have the potential to cover the entire city at a reasonable cost.

The parallel growth of computation power and data volume kept in check the tradeoff between computational capacity and classification performance. On the one hand, Computation Processing Units (CPU) and Graphical Processing Units (GPU) have created sizeable extra computation potential. On the other hand, the pursuit of better accuracy leveraging, for example, the pervasive introduction of cheap sensors and rich Geographic Information Systems (GIS), immediately absorbed this additional capacity. Overall, transportation mode classifiers deployed on data from urban and densely populated areas did not increase their performance proportionally with the data consumption. Therefore, statistical methods developed before the Big Data paradigm [27], and machine learning methods developed after [18], may still compete. A factor emerging from the literature is that methods still depend

heavily on labels. Even though some semi-supervised configuration of artificial neural networks exists in this field and reduces the need for labels in the classifier's training phase, filtering a subset of high-quality labels from Big dataset is still very challenging and hardly scalable. For example, continuous disruptions of transport operations due to roadwork or special events would also disrupt any classifier trained with labels that no longer reflect the transport network [28]. Even in the assumption of operations stability, the impact of flipping and overlaying labels–potentially present due to human collection errors–seems still critical. Supervised classifiers deployed on time series, e.g., for the BIBO task, could deliver biased classifications and threaten the system's sustainability at scale. The problem deserves more attention in this field, and for time series requires at least the same attention granted to independent and identically distributed data. Systematic studies and appropriate methodologies in the second case exist, such as for image classification. However, for time series classification these contributions are only partially applicable. Furthermore, existing preliminary studies about the impact of flipping labels on time series classification show that severe bias on the measurements of these classifiers' performance is present when just 10% of the labels are wrong. In such a case, although the classifiers might be resilient to labels' noise, analysts and practitioners would base their decisions on a biased performance evaluation, simply because the error rate in human validated labels is unknown [13].

1.2. Contribution of the Paper

This paper focuses on the combined use of GPS and BLE signals for unsupervised autovalidated BIBO classification of bus passengers. Representing the user via the smartphone and the bus via a BLE device, we use sensors signals as pseudo labels to learn discriminating when a user is inside (BI) or outside (BO) the bus.

The central intuition is that when the user is inside the bus (BI) the distance between smartphone and bus should be close to zero, and the proximity to BLE devices installed in the bus would cause the highest signal strength. Vice-versa, when the user is outside the bus (BO), the considerable distance between the user and the BLE device should cause the lowest signal strength or no signal at all.

To learn the cause-effect relationship between smartphone-bus proximity and BLE signal strength, we implement two parallel Wasserstain Autoencoders (WAE). One learns how to reconstruct the time series of the BLE sig-

nal (effect) given the smartphone-bus proximity (cause). Given the BLE signal strength (effect), the other learns to rebuild the smartphone-bus distance (cause). We define this configuration as a cause-effect multi-task Wasserstein Auto-encoder (CEMWA). From the unsupervised training of this CEMWA, we learn to reduce the description of the interaction between passengers and buses to only four dimensions. In this 4-dimensional latent space, the observations self-organize such that discrimination between BI and BO classes is possible through unsupervised clustering with Density-based spatial clustering of applications with noise (DBSCAN).

CEMWA combines and extends the following frameworks. (i) Split-brain Auto-encoder configuration by Zhang et al. [29]; (ii) Deep clustering for unsupervised learning by Caron et al. [30]; (iii) Multi-task formulation of the objective function by Kendall et al. [31]; (iv) Maximum Mean Discrepancy (MMD) formulation of the objective function for generative models by Gretton et al. [32]; and (v) MMD extension to Wasserstein Auto-encoders by Tolstikhin et al. [33].

The resulting architecture solves the scalability problem related to noise in labels. We perform an ablation study including traditional WAE architectures and supervised methods. Results show that our unsupervised classifier solves the negative impact of the label-induced bias affecting supervised classifiers. Moreover, the architecture we propose embodies a solution for signal data imputation, which is generally a critical and separate step necessary to perform good classification. Finally, since the method relies only on the interaction between smartphone and bus, temporary or permanent disruptions of the network would not affect the classification task.

2. Methods and Materials

This section presents a number of frameworks supporting our goal of substituting ordinary labels for training supervised or semi-supervised artificial neural networks specialized in processing GPS signal. Three are the main steps behind the intuition. Firstly, instead of labels we leverage an independent sensor time-series–BLE–for representation learning of cause-effect relationship between GPS and BLE. Secondly, to avoid confounding correlations between the two sensors' signals, we design and fine-tune a specific encoder-decoder architecture based on a general formulation of regularized auto-encoders. Lastly, with DBSCAN, we turn into classes the representations learned via independent sensors time-series–GPS and BLE.

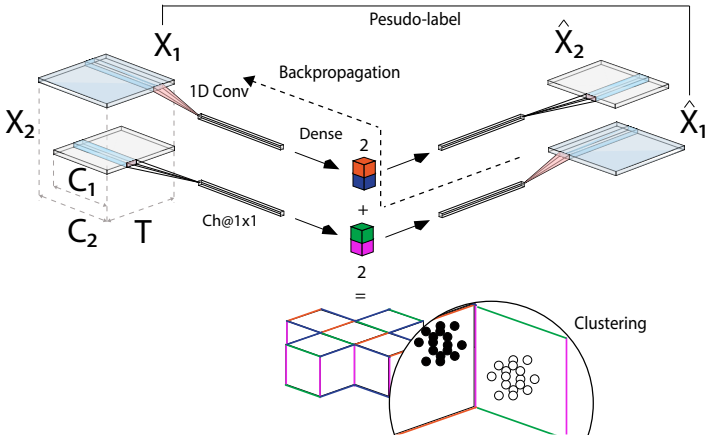

Figure 1: Cause-effect Multi-task Wasserstein Auto-encoder (CEMWA) independent cross-reconstruction of X_1, X_2 minimizing (7) and clustering of the resulting latent space, 5028 parameters.

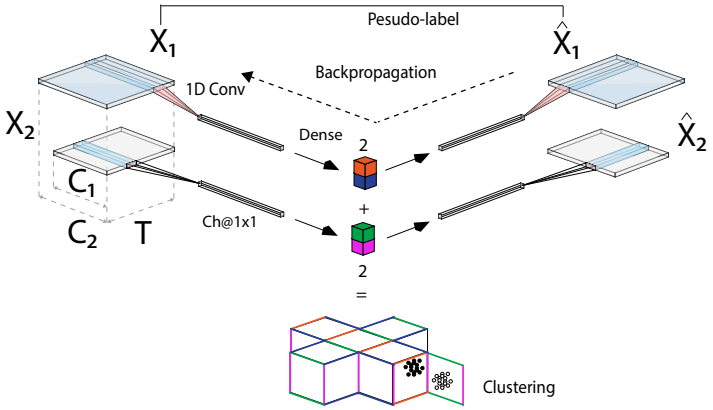

Figure 2: Multi-task Wasserstein Auto-encoder (MWA) independent reconstruction of (X_1, X_2) minimizing (3), with $c = \mathcal{L}_{WAE}$ and clustering of the resulting latent space, 5028 parameters.

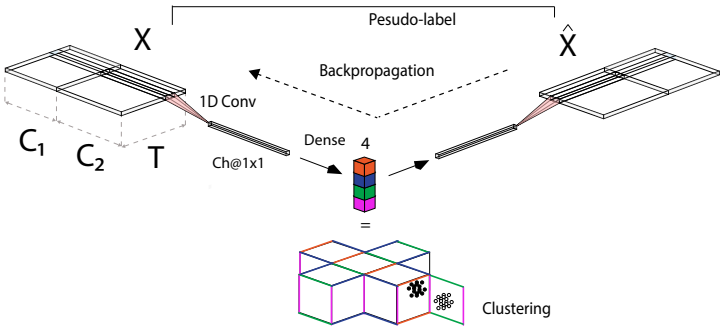

Figure 3: Wasserstein Auto-encoder (WA) reconstruction of $X = (X_1, X_2)$ minimizing (1) and clustering of the resulting latent space, 4932 parameters.

Following the notation of Tolstikhin et al. [33], we identify sets with calligraphic letters (i.e. $\mathcal{X}$), random variables with capital letters (i.e. X)., and values with lower case letters (i.e. x).

Let $X \in \mathbb{R}^{t \times d}$ be the tensor describing the smartphone/bus interaction, in a time window of t observations, which d independent feature channels express such that: $X_1 \in \mathbb{R}^{t \times d_1}$ represents the channels deriving from the GPS sensors; $X_2 \in \mathbb{R}^{t \times d_2}$, from the BLE devices network; where $(X_1, X_2) = X$ and $\mathcal{D}_1 \bigcup \mathcal{D}_2 \subseteq \mathcal{D}$, with $|\mathcal{D}| = d$.

We would like to learn a representation for X solving the prediction problem $\hat{X} = (\hat{X_1}, \hat{X_2})$, where $\hat{X_1} = \mathcal{F}_1(X_2)$, and $\hat{X_2} = \mathcal{F}_2(X_1)$. $\mathcal{F}_1$ learns the cause-effect relationship between smartphone-bus proximity and BLE signal strength, while $\mathcal{F}_2$ learns the inverse cause-effect relationship of the same interaction between smartphone and bus.

$\mathcal{F}$ represents a class of non-random generative Encoder/Decoder models determinalistically mapping input points to the latent space with a convolutional neural network (CNN) via Encoder, and latent codes to output points with a transpose CNN via Decoder. To learn $\mathcal{F}$, we minimize the Wasserstein optimal transport cost (1) between the true-unknown data distribution P_X and the latent variable model P_G specified by the prior distribution P_Z of latent codes $Z \in \mathcal{Z}$ and the generative model $P_G(X|Z)$ of the data points

$X \in \mathcal{X}$ given Z [33]. (1) shows that while the decoder pursues the encoded training examples reconstruction at the minimal cost c, the encoder pursues two conflicting goals at the same time: (i) Match the encoded distribution Q_Z to the prior distribution P_Z, where $Q_Z := \mathbb{E}_{P_X}[Q(Z|X)]$ (ii) Ensure that the latent representation for the decoder allows accurate reconstruction of the encoded training examples. In this two steps procedure, first Z is sampled from a fixed distribution P_Z on a latent space $\mathcal{Z}$, and then Z is mapped to $\hat{X} = G(Z)$ for a given map $G : \mathcal{Z} \to \mathcal{X}$, where $\hat{X} \in \mathcal{X} = \mathbb{R}^{t \times d}$.

$$\begin{aligned} \mathcal{L}_{WAE}(P_X, P_G) := & \inf_{Q(Z|X) \in \mathcal{Q}} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)} \left[c(X, G(Z)) \right] \\ & + \lambda \cdot \mathcal{D}_Z(Q_Z, P_Z), \\ & \lambda > 0 \end{aligned} \tag{1}$$

This task formulation extends the Split-brain Autoencoder proposed by Zahng [29]. We share the intuition, and the goal of achieving a representation containing high-level abstraction and semantics of the smartphone-bus interaction registered independently by GPS and BLE sensors. In contrast with Zahng, we aim at learning the cause-effect function and its inverse, separately, and not just merely as a "pretext". However, to keep up with the Big Data scale, Zhang approach brings some limitations with the objective function in Eq. (2): (i) For weighting the multi-task cost $\mathcal{O}$, Zhang introduces the hyperparameter $\hat{\lambda}$ that requires a dedicated optimization process. (ii) To learn cause-effect relationship and its inverse, we do not want include the full signal $c((\mathcal{F}_1(X_2), \mathcal{F}_2(X_1)), X)$ in the multi-task objective function $\mathcal{O}$. (iii) The use of a classical unregularized auto-encoder, which minimizes only the reconstruction cost c, between X and $\hat{X}$, prevents from yielding full advantage of representation learning for this problem, facilitating model over-fitting instead of generalization power.

$$\begin{aligned} \mathcal{O} = arg \min_{\mathcal{F}_1, \mathcal{F}_2 \in \mathcal{F}} [& \hat{\lambda} \cdot c(\mathcal{F}_2(X_1), X_2) \\ & + \hat{\lambda} \cdot c(\mathcal{F}_1(X_2), X_1) \\ & + (1 - 2 \cdot \hat{\lambda}) \cdot c((\mathcal{F}_1(X_2), \mathcal{F}_2(X_1)), X)], \\ & \hat{\lambda} \in [0, \frac{1}{2}] \end{aligned} \tag{2}$$

In the following sections we can now look at how we extended Zhang's work to cover both of the aforementioned limitations and enable clustering.

2.1. Extension Towards Multi-task Self-learned Cost Weights

In a multi-task setting, Kendall shows that when tasks uncertainty depends on its unit of measure, homoscedastic uncertainty is an effective bias for weighting multiple losses [31]. This fits exactly with our problem, where the proximity between smartphone and bus is measured in meters on one hand, and in Received Signal Strength Indicator (RSSI) on the other hand. With $\hat{X}_1 = \mathcal{F}_1(X_2)$ and $\hat{X}_2 = \mathcal{F}_2(X_1)$, where $\mathcal{F}_1, \mathcal{F}_2 \in \mathcal{F}$, (3) represents the multi-task loss formulation for our problem, according to Kendall. The main difference between (2) and (3) is that in the second case the two parameters can be "learned" leveraging the ANN back propagation algorithm while learning $\mathcal{F}$ parameters, during the training phase. When training on large datasets, this is an advantage.

$$\begin{aligned}\mathcal{O} = arg\min_{c}[&\frac{1}{2\sigma_1^2} \cdot c(\hat{X}_1, X_1) \\ &+ \frac{1}{2\sigma_2^2} \cdot c(\hat{X}_2, X_2) \\ &+ \ln \sigma_1 + \ln \sigma_2]\end{aligned} \tag{3}$$

2.2. Extension towards regularized auto-encoder

WAE represent a class of generative models resting on the optimal transport cost derived from [34] and expressed in (1). This class underpins our extension: In contrast to Zhang work [29], which studies the unregularized cost c, such as regression and cross-entropy, we include to the regression cost a regularization term, i.e., the maximum mean discrepancy (MMD) $D_Z = MMD_k(P_Z, Q_z)$. (4) expresses the MMD, where $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a positive-definite reproducing kernel, and $\mathcal{H}_k$ is the reproducing kernel Hilbert space (RKHS) of real-valued functions mapping $\mathcal{Z}$ to $\mathbb{R}$ [32].

Similarly to variational auto-encoders (VAE) [35], this WAE-MMD formulation uses artificial neural networks (ANN) to parametrize encoder and decoder. However, to allow back-propagation throughout decoder and encoder, the re-parametrization trick [35] *"forces $Q(Z|X = x)$ to match P_Z for all the different samples x drawn from P_X. In contrast, WAE forces the continuous mixture $Q_Z := \int Q(Z|X)dP_X$ to match P_Z"* [33]. Consequently, WAE allow a better organization of the latent space which we leverage for clustering. Compared to alternative formulations of the penalty term, such as the Generative Adversarial Networks [36] (GAN), or in general the WAE-GAN [33], where $\mathcal{D}_Z$ in (1) is the Jensen-Shannon Divergence, the literature

shows slightly better reconstruction performance for $\hat{X}$ but at the heavy cost of an additional network and possibly complex and multi-modal distributions for P_Z. Since our problem is simple in principle, we opt for simplicity, thus for MMD.

$$\begin{aligned} MMD_k(P_Z, Q_z) = || \int_{\mathcal{Z}} k(z, \cdot) dP_Z(z) \\ - \int_{\mathcal{Z}} k(z, \cdot) dQ_Z(z) ||_{\mathcal{H}_k}, \end{aligned} \tag{4}$$

If k is characteristic[1] MMD represents a divergence measure [37].

We try both the alternative kernels k proposed for Wasserstein auto-encoders (WAE) [33]: Radial basis function kernel (RBF) (5); and Inverse multiquadratics kernel (6).

$$k^{RBF}(z, \tilde{z}) = e^{\frac{-||\tilde{z}-z||_2^2}{\sigma_k^2}} \tag{5}$$

$$k^{IMK}(z, \tilde{z}) = \frac{C}{C + ||z - \tilde{z}||_2^2} \tag{6}$$

The resulting architecture consists of two independent encoder/decoder maps $\mathcal{F}_1, \mathcal{F}_2 \in \mathcal{F}$ such that $\hat{X}_1 = \mathcal{F}_1(X_2)$ and $\hat{X}_2 = \mathcal{F}_2(X_1)$. Each map's encoder consists of 1D-Convolutions; 1D-Transpose-Convolutions for the decoder. As described in Fig. 1, maps are learned using back-propagation to minimizing the multitask formulation of our objective function (7), where we set $c = ||X - \hat{X}||_2^2$ and $D_Z = MMD_k$. To find optimal relative weights between tasks, we leverage the same back-propagation algorithm.

$$\begin{aligned} \mathcal{O}_{WAE} = arg \min_{\mathcal{F}_1, \mathcal{F}_2 \in \mathcal{F}} & \frac{1}{2\sigma_1^2} \cdot \mathcal{L}_{WAE}(\mathcal{F}_2(X_1), X_2) \\ & + \frac{1}{2\sigma_2^2} \cdot \mathcal{L}_{WAE}(\mathcal{F}_1(X_2), X_1) \\ & + \ln \sigma_1 + \ln \sigma_2 \end{aligned} \tag{7}$$

[1]Given $k : \mathcal{Z}^+ \to \mathbb{R}$, k is injective, $\mathcal{Z}^+$ is positive and represents the set of probability measures on $\mathcal{Z}^+$

2.3. Extension of Deep Clustering Architecture

To allow unsupervised classification of images, Caron et al. proposes a straight ANN predicting cluster assignment as pseudo-labels [30], and iterate between clustering with k-means [38] and back-propagation to update the network's weights after the cluster assignment. The intuition is that clustering provides and alternative and meaningful reference to labels. Therefore, the loss function is computed against clusters instead of known labels. However, since we collect two independent measure of the same event, by design, we tweak the process using these two signal as reciprocal pseudo-labels instead. When back-propagation converges, we perform clustering of data representation on the latent space with DBSCAN [39]. Fig. 1, 2 and 3 show the architectures tested within our ablation study: the first leverages the known cause-effect relationship between GPS and BLE signal; the second, the multi-task independent reconstruction of the two signals; the last shares parameters within the same network, to reconstruct a tensor where multiple channels contain each available signal.

2.4. Final Model Formulation

Fig. 1 presents the final structure of our CEMWA model, resulting from the Split-brain's architecture extensions described in Sec. 2.1, 2.2 and 2.3.

We will argue as follows: (i) CEMWA has the ability of learning the cause-effect relationship between GPS and BLE signals recording smartphone-bus interactions. (ii) Learning such a relationship allows the exposure of self-validated features characterizing the BIBO status of users with respect to buses. (iii) These self-validated features allow unsupervised classification of users trajectories, where smartphones identify users and BLE devices identify buses. (iv) Alternative unsupervised architectures leveraging the correlation instead of cause/effect between the GPS and BLE signals—such as those described in Fig. 2 and 3— are unable to to perform self-validated unsupervised BIBO classification. (v) In case of labels noise, CEMWA significantly outperforms the most accurate supervised classifiers, such as random forest or XG-boost (extreme gradient boosting). (vi) Regardless of the classification performance, CEMWA embodies both a data imputation and a validation mechanism, while supervised classifiers or alternative unsupervised architectures should rely on dedicated processes, such as an exponential weighted moving average for BLE or GPS imputation [40], and user validation for BIBO labels [13, 6].

To substantiate our hypotheses through the following experiments, consistently, we designed and deployed a specific sensing architecture, and collected high quality ground truth.

2.4.1. Ground truth collection, data cleansing, and preparation

CEMWA's architecture mirrors the smartphone sensing platform we designed and deployed to track the activity of three autonomous buses operating an experimental public service in Denmark, between two extremes of the Lyngby campus where the Technical University of Denmark is located.

During operations these buses are tracked via GPS available from the bus telemetry, while test passengers recruited for the experiment are tracked via smartphones. The sensing platform collected GPS signals that both smartphones and buses generate. GPS collection was strictly limited around the operations area using a geo-fence [41]. In the same area, we deployed 300 BLE devices: one on each bus and bus stop, plus one at the entrance/s of each building in the campus.

To become a test passenger, each user provided explicit agreement to terms and conditions presented in compliance with the General Data Protection Regulation[2]. The sensing platform supports both Android and iOS devices, and the Apps are published on GooglePlay[3] and App Store[4] respectively. This project is a social science study, includes data and numbers only, is not a health science project, and does not include human biological material nor medical devices. Consequently, in Denmark, where the data collection took place, the Health Research Ethics Act provides a dispensation for notification to any research ethics committee.

When the smartphone is within the relevant geo-fence, in optimal conditions, the platform collects GPS with 1 s resolution. Simultaneously, with the same resolution, the platform samples RSSI signal strength of BLE devices "visible" in the range of each smartphone.

We extracted the trajectories of both test passengers and buses between 1st April and 1st July. 134 users generated a total of $4,584,000$ GPS observations; three buses, $1,162,000$ GPS observations, for a total of approximately $940\,\mathrm{h} \cdot$ bus operations (see Fig. 7).

From the remaining set of data we extracted the sub-set of observations

[2]Information provided to users before recruitement, access on 03-09-2021

[3]LINC DTU at GooglePLay, access on 03-09-2021

[4]LINC DTU at Appstore, access on 03-09-2021

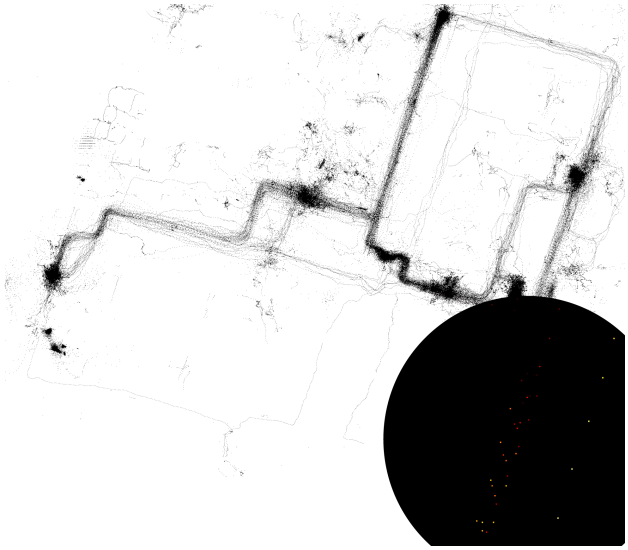

Figure 4: Subset of GPS points presenting at least one BLE device reading; color map based on e^{speed} shows that buses and other modes in the area have the same speed distribution– i.e., walk and bike–few trajectories recorded from car are the only exception.

Figure 5: GPS points from smartphones, color map based on spatial density shows bus stops and bus deposit.

Figure 6: Subset of GPS points presenting at least one BLE device reading; points spatial distribution shows higher density at the bus stops, bus deposit and some buildings.

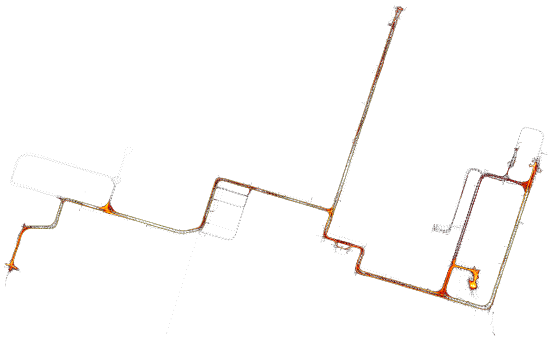

Figure 7: GPS points from buses, spatial distribution shows higher density at the bus stops, bus deposit.

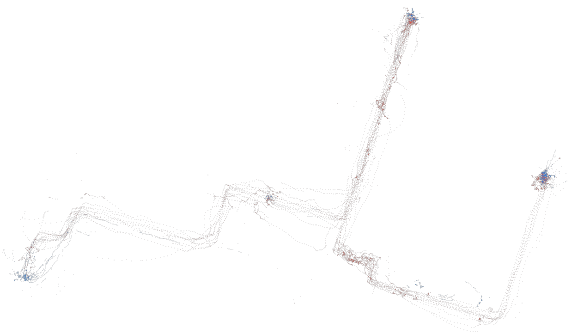

Figure 8: Be-In (BI) clusters identified on smartphone data clustering CEMWA latent space with DBSCAN, and colored with ground truth labels. Red color depicts users inside the bus; blue color, users outside the bus.

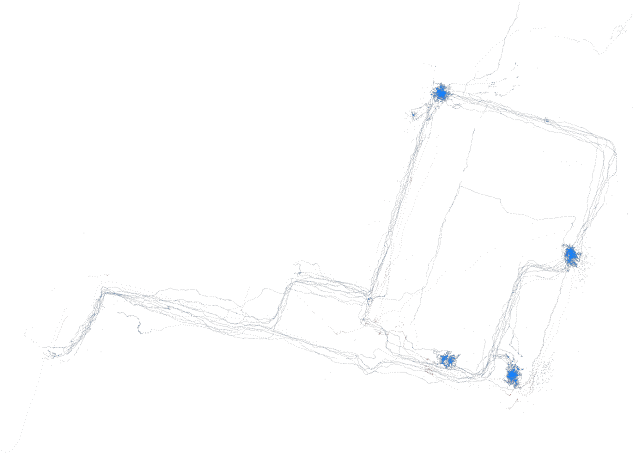

Figure 9: Be-Out (BO) clusters identified on smartphone data clustering CEMWA latent space with DBSCAN, and colored with ground truth labels. Red color depicts users inside the bus; blue color, users outside the bus.

containing at least one BLE observation, for a total of 195,000 GPS observations (see Fig. 6). This set present the maximum BLE resolution available, while the corresponding GPS resolution is below the maximum resolution available within the dataset. No labels are available for this set. Fig. 4 depicts the speed distribution of different transportation modes present in this subset. To highlight the differences in speed between different transport mode, we applied the exponential transformation. However, the black flat color shows that the speed distribution seems to be the same in all the cases, except for some cars (see black magnified detail).

Outside the passengers' set, we generated a set of records counting 59,000 observations which are part of a specific experiment where seven components of the project's staff collected via smartphone a high quality BIBO labels and observations set (see Fig. 5), following the same methodology of Shankari et al. for MobilityNet dataset collection [14]. Thus, to avoid bias in the labels, we provided instructions on precise origin-destination sequences, divided in three different trip-groups. Each staff member has been randomly assigned to a trip-group. After watch synchronization, during the experiment, each staff member annotated the hour and minute each time s/he boarded or alighted a bus.

2.4.2. Experiment setup

Table 1 describes experimental setup for the evaluation of supervised baselines, for ablation study of various unsupervised architectures, and for the model we propose in this work. We applied a trajectory segmentation considering each pair of points beyond 120 s time-range, or where the space variation over time variation is beyond 120 m/s, the end of a segment and the beginning of the next segment. After segmentation, for each segment we applied a sliding window including 9 consecutive points and 1 step stride. CEMWA, MWA and WA process the resulting tensor straightly, using convolutions. Instead, Random Forest and XGboost require an intermediate process to extract traditional features from the 9 step windows contained in each segment, computed at each slide, applying the same stride of 1 step.

We setup the same conditions for both baselines and proposed methods. Comparing supervised and unsupervised classifiers in this setting is subject to the limitation of labeled dataset. As we want to provide performance distributions instead of points, with supervised methods we apply leave-one-out validation method, while with the unsupervised methods we apply a hold out method. In the first case we train the model with all the users belonging

to the labeled observations except one, which represent the test set. In the test set we rotate all the users available. Thus, the main scores can presented as mean ± standard deviation. In the second case, we train the model with the unlabeled observations, and without performing DBSCAN clustering. Then we use the model including DBSCAN to classify—off the sample—the labeled observations. Similarly, we can present the main scores as mean ± standard deviation. Consequently, we can compare these scores even though the training process is quite different.

This setup assumes that the ground truth quality is stable and high. As we mentioned, the labels collection method we used can guarantee a higher quality level on the labels. Unlike the case where ground truth is collected from passengers, the project's staff followed instructions and was not subject to, e.g., recall bias, and less likely to suffer systematic and random distractions. Therefore, to provide an exhaustive picture for performance, we train these supervised methods adding some noise in the training set, i.e., flipping a controlled percentage of labels. We sample the number of errors per user from a Poisson distribution and we flip labels accordingly. The test set is not affected. Therefore, applying a Monte Carlo evaluation based on 100 loops per experiment, and on the same setup described in Table 1, we can estimate the sensitivity to labels noise. This problem does not affect the unsupervised methods, which use Bluetooth RSSI signal as pseudo-labels instead (see Table 1, Signals row).

3. Results and Discussion

After a manual optimization process of CEMWA, MWA, and WA, we yield optimal performance with the combination of hyperparameters described in Table 2. As opposed to CEMWA, MWA and WA converge to a relatively lower loss, and overfitting is higher. Although the three models have the same number of parameters, we record differing computation times for the training phase (which might be justified by concurrent processing on GPU). Compared to MWA and WA, CEMWA achieves substantially better scores, with higher mean and inferior standard deviation. (5) yields the results we present, while (6) seems not effective in this use case. We apply the same penalization across all three models during back-propagation to rebalance BI and BO classes when computing the WAE loss within the optimizer. Rather than the Precision score, the Recall score of the BI class seems

Table 1: Experiment Setup

	Supervised Baseline **XG-Boost** **Random Forest**	**Unsupervised Baseline** **MWA (Fig. 2)** **WA (Fig. 3)**	**CEMWA** (Fig. 1)
Smartphone Set **GPS + BLE** **Android + iOS**	59,000 labelled observations 7 users	328,000 tot observations 59,000 labelled 134 tot users	
Buses set	1,162,000 observations, 940 h · bus, 3 buses		
Signals	Speed, Longitude, Latitude, Timestamp from GPS	Speed, Longitude, Latitude, Timestamp from GPS RSSI and Timestamp from BLE devices	
Use of Ground Truth Labels	For training and evaluation	For evaluation only	
GPS Trajectory Segmentation	time gap between points >120 s determines a new segment points representing speed >45 m/s determine a new segment		
Data Cleansing	Segments <10 consecutive points are discarded		
Observation Imputation	Imputation with Exponential Weighted Moving Average and Masking	Masking Only	
Basic Feature Extraction	time-, space-gap, and bearing between each pair of GPS points, GPS distance between smartphone and buses within 1 s range		
Time Series Sliding Window	moving window of 9 consecutive steps segment, and 1 step stride		
Feature Extraction **on Sliding Window**	Mean value Max value Min value Position of the minimum value Position of the maximum value Amplitude between min and max value Number of points beyond one std dev. Number of points below one std dev. Number of points above one std dev. Number of peaks in the mvoing window Number of peaks half sliding window Number of peaks above 1 one std dev. Peak distance within sliding window Slope	None. ANN performs features extraction. Encoder, 1 convolutional neural network. Decoder, 1 transposed convolutional neural network. Convolution Kernel: 3 $\lambda \in [10^{-4}, 1]$ Batch Size: $\in [16, 1024]$ true sample size: $\in [10, 100]$ Learning Rate: $\in [10^{-5}, 10^{-1}]$ Epochs: $\in [10, 100]$	
Performance Evaluation Method	Leave-one-out: One user in the test-set Training-set is the complementar set. Repeated rotating each user in test-set.	Hold-out: Training- and validation-set from unlabelled-set. Test-set corresponding to the labelled-set.	
Method performance distribution	Given by performance on individual users of whole the labelled set.		
Performance Metric	AUC ROC, F1-score, Precision, Recall, Accuracy		

to provide an essential contribution to the overall superior performance of CEMWA.

The supervised methods we evaluate are performing very well. XGboost presents a slightly higher score than CEMWA but with a slightly larger standard deviation. The two models seem to have comparable performance in terms of computation time. There seems to be the following differences. In optimal conditions and ground truth quality, XGboost appears to record a substantially higher precision score, but a lower recall score than CEMWA. Under the same conditions, Random Forest seems comparable with MWA and WA, or better. But we should not forget the impact of wrong labels in the training process of supervised methods such as XGboost and Random Forest. This problem does not affect unsupervised methods like CEMWA.

To test the sensitivity of XGboost and Random Forest to noise in the labels, we run a Monte Carlo evaluation. Results show that beyond 10% flipped labels during training leads to substantial performance degradation. This rapid degradation is of critical importance when labels are collected

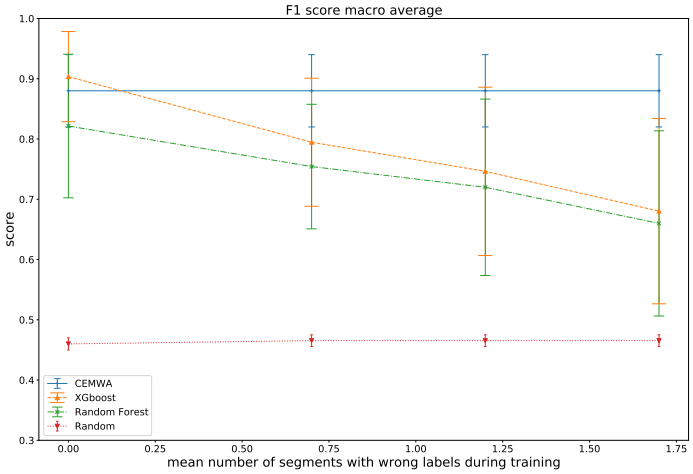

Figure 10: Impact of wrong labels on supervised classifiers training (F1 score macro average).

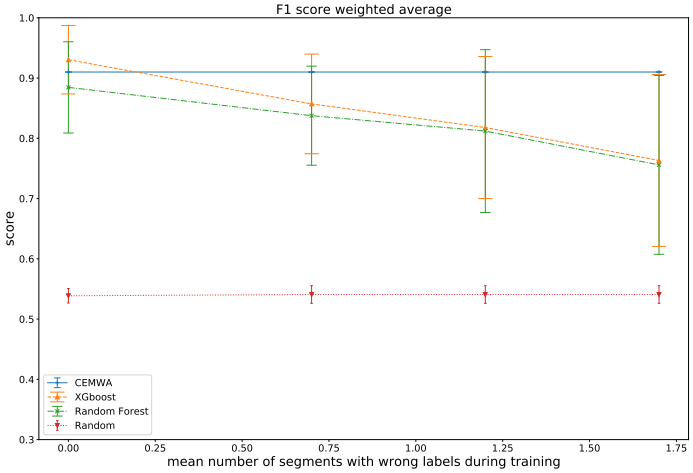

Figure 11: Impact of wrong labels on supervised classifiers training (F1 score weighted average).

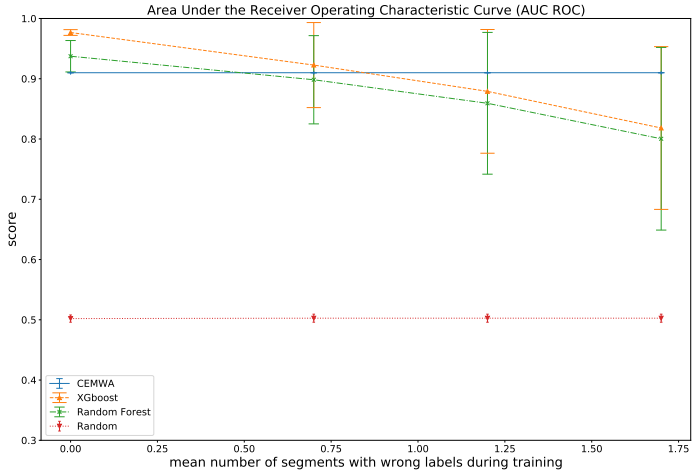

Figure 12: Impact of wrong labels on supervised classifiers training (AUC ROC).

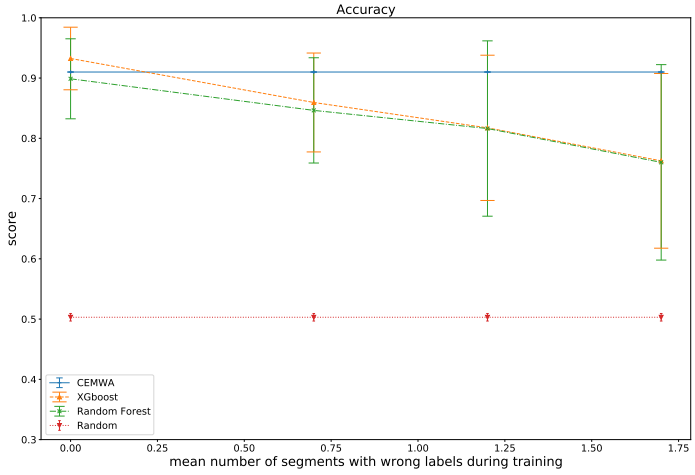

Figure 13: Impact of wrong labels on supervised classifiers training (Accuracy).

Table 2: Encoder/Decoder CNN architecture hyperparameters, final configuration for CEMWA, EMWA, and WA.

Encoder	
Convolutional Neural Network (CNN) Layers	1
Activation Function	Rectified Linear Unit
Fully connected Layers	0
Dropout	0.25
Decoder	
Transposed CNN Layers	1
Activation Function	Leaky Rectified Linear Unit
Fully connected Layers	0
Dropout	0.25
Optimizer	Adam
Epochs	50
Batch Size	32
Learning Rate	$10^{-}4$
Dropout	0.25

directly from passengers. Consequently, the trade-off between the cost and the quality of labels collection critically impacts the scalability potential of supervised methods. Figure 10 depicts the impact of wrong labels on the classifiers performance: When users provide wrong labels to less than 1 segment in average–where a segment is defined according to the GPS Trajectory Segmentation of Table 1–the performance of supervised classifiers drops dramatically compared to CEMWA.

This configuration provides potential for enhancing smartphone battery efficiency and user privacy, because: (i) Smartphones would listen to Bluetooth, while keeping GPS up, with minimum resolution, just enough to avoid GPS cold start; (ii) Bluetooth in proximity would trigger higher resolution GPS, only when necessary.

In practice, after cause-effect training with encoder-decoder architecture and clustering–where GPS compression is trained reconstructing BLE and vice-versa–CEMWA could be deployed as follows. During operations, one CEMWA's encoder compresses GPS, while a separate encoder compress Bluetooth. The two independent compressed representation are joined into one.

The proximity between the resulting representation and the clusters determine whether the observation belong to BI or BO class.

For applications where disruptions are unlikely–thus we expect a stable process in time–the amortization of high-quality ground truth could rely on a longer time horizon. An established metro line for example, is unlikely to experience changes frequently. In contrast, bus services are subject to continuous disruptions, e.g., roadworks and traffic congestion. Therefore, a supervised BIBO classifier could be a good choice in the first case. However, the unsupervised BIBO classifier seems better in the second case. Results rely mainly on the smartphone-bus-distance. This feature can be challenging to compute off-line, especially when a large number of passengers and vehicles are active. However, a federated-learning design [42] would solve the problem, and allow the computation of features online.

Assuming smartphones' future market penetration stable, and relying on adversarial sensors architectures, we show an approach to substitute manually collectible labels. This approach has vast potential; for example, BLE beacons contraposed to GPS within a CEMWA architecture would enable ticketless transit across any public transportation system, and large-scale deployment, even for applications subject to frequent disruptions. In addition to the before-mentioned use case, we suggest road and bridge tolls or sharing mobility services like cars, bikes, or scooters. A BIBO system also supports visually impaired people to chose to board the right bus from the bus stop or to alight at the right stop from the bus. It could facilitate the integration across multiple service providers, operating mostly on software instead of physical infrastructure, even integrating with existing CICO and WIWO systems.

Table 3: Results with optimal Ground Truth for method evaluation and training of supervised algorithms

Model	Task	Labeled Observations	Unlabeled Observations	Precision	Recall	F1-score		Accuracy	AUC ROC	Model Parameters	Computation-time		Evaluation Method
						macro average	weighted average				Training	Feature Extraction	
CEMWA	BI	13,154	191,556	0.77	0.89	0.88 ± 0.06	0.92 ± 0.04	0.91 ± 0.04	0.91	5028	97 min on 191,556 set	< 1 min	Hold out (scrores distribution on labeled-set, comparable with leave-one-out)
	BO	45,327		0.97	0.92								
MWA	BI	13,154	191,556	0.52	0.66	0.72 ± 0.26	0.79 ± 0.22	0.79 ± 0.26	0.72	5028	47 min on 191,556 set		
	BO	45,327		0.89	0.82								
WA	BI	13,154	191,556	0.76	0.53	0.77 ± 0.15	0.85 ± 0.08	0.86 ± 0.08	0.74	4932	23 min on 191,556 set		
	BO	45,327		0.87	0.95								
XG-boost	BI	13,154	not applicable	0.84	0.78	0.90 ± 0.07	0.93 ± 0.06	0.93 ± 0.05	0.98	not applicable	< 1 min	31 min on 58,481 set	Leave-one-out
	BO	45,327		0.93	0.95								
Random Forest	BI	13,154		0.90	0.43	0.82 ± 0.11	0.88 ± 0.07	0.90 ± 0.05	0.90				
	BO	45,327		0.85	0.99								
Random Classifier	BI	13,154		0.24	0.50	0.46 ± 0.01	0.54 ± 0.01	0.50 ± 0.003	0.50		not applicable	not applicable	not applicable
	BO	45,327		0.76	0.50								

4. Conclusion

This paper focuses on an implicit tracking system to detect whether a passenger is inside or outside the transport network. To avoid using labels in the classifier training, we leverage a novel artificial neural network architecture learning the cause-effect relationship between two independent sensors measuring the same event. We call this approach CEMWA. In optimal conditions and with high-quality ground truth, CEMWA's performance is comparable or better than both supervised and unsupervised baselines. CEMWA and XGboost performance evaluated with optimal knwoledge on BIBO ground truth seem promising for public transport ticketing in general. In situations with noisy ground truth–such as transport services subject to disruption or surveys where passengers lack the ticket payment as an incentive to provide exact ground truth–we show that supervised classifiers' performance degrades. Supervised methods' tolerance to noisy labels is case specific. However, the issue does not affect CEMWA by design. Consequently, this unsupervised method is both scalable and fulfills the requirements for use-cases where, e.g., frequent service disruptions may lead to the need for regular labels' collection. Future research will investigate in few directions: (i) The extension of a sensor-to-sensor validation on new signals and neural network architectures, the sensitivity to labeling noise; (ii) The introduction of sensitivity to noise as a performance index to evaluate and compare supervised methods; and (iii) The connection between dry machine learning scores of our BIBO classifier and key performance index assessing automatic fare collection systems with BIBO.

Acknowledgment

This project is co-financed by the European Regional Development Fund through the Urban Innovative Actions Initiative.

References

[1] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselbock, N. Hofler, Be-in/be-out with bluetooth low energy: Implicit ticketing for public transportation systems, Ieee Conference on Intelligent Transportation Systems, Proceedings, Itsc 2015- (2015) 7313345. doi:10.1109/ITSC.2015.253.

[2] S. Hietanen, Mobility as a service, the new transport model 12 (2014) 2–4.

[3] D. A. Hensher, C. Mulley, Hensher, d.a. and mulley, c. mobility bundling and cultural tribalism - might passenger mobility plans through maas remain niche or are they truly scalable?, Transport Policy 100 (2021) 172–175. URL: https://www.sciencedirect.com/science/article/pii/S0967070X20309203. doi:https://doi.org/10.1016/j.tranpol.2020.11.003.

[4] W. Goodall, T. Dovey, J. Bornstein, B. Bonthron, The rise of mobility as a service, Deloitte Rev 20 (2017) 112–129.

[5] M. H. Wirtz, J. A. Klähr, Smartphone based in/out ticketing systems: A new generation of ticketing in public transport and its performance testing, Wit Transactions on the Built Environment 182 (2019) 351–359. doi:10.2495/UT180321.

[6] V. Servizi, C. F. Pereira, K. M. Anderson, A. O. Nielsen, Transport behavior-mining from smartphones: a review., European Transport Research Review (2021). URL: https://doi.org/10.1186/s12544-021-00516-z. doi:10.1186/s12544-021-00516-z.

[7] O. Baescu, H. Christiansen, The Danish National Travel Survey Annual Statistical Report TU0619v2, DTU Management, 2020. doi:10.11581/dtu:00000034.

[8] Y. Cui, S. S. Ge, Autonomous vehicle positioning with gps in urban canyon environments, Ieee Transactions on Robotics and Automation 19 (2003) 15–25. doi:10.1109/TRA.2002.807557.

[9] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, D. Roggen, Enabling reproducible research in sensor-based transportation mode recognition with the sussex-huawei dataset, IEEE Access (2019). doi:10.1109/ACCESS.2019.2890793.

[10] E. Foxlin, Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter, in: Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium, 1996, pp. 185–194. doi:10.1109/VRAIS.1996.490527.

[11] A. Bjerre-Nielsen, K. Minor, P. Sapiezynski, S. Lehmann, D. D. Lassen, Inferring transportation mode from smartphone sensors: Evaluating the potential of wi-fi and bluetooth, Plos One 15 (2020). doi:10.1371/journal.pone.023400310.1371/journal.pone.0234003.r00110.1371/journal.pone.0234003.r00210.1371/journal.pone.0234003.r00310.1371/journal.pone.0234003.r00410.1371/journal.pone.0234003.r005.

[12] I. Malmberg, An analysis of iBeacons and critical minimum distances in device placement, Ph.D. thesis, 2014. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-187925.

[13] V. Servizi, D. R. Persson, P. Bækgaard, H. Villadsen, I. Peled, J. Rich, F. C. Pereira, O. A. Nielsen, Context-aware sensing and implicit ground truth collection: Building a foundation for event triggered surveys on autonomous shuttles: Artikel, in: Proceedings from the Annual Transport Conference at Aalborg University, volume 28, 2021.

[14] K. Shankari, J. Fuerst, M. F. Argerich, E. Avramidis, J. Zhang, Mobilitynet: Towards apublic dataset for multi-modal mobility research, Climate Change AI (2020).

[15] V. Servizi, N. C. Petersen, F. C. Pereira, O. A. Nielsen, Stop detection for smartphone-based travel surveys using geo-spatial context and artificial neural networks, Transportation Research Part C: Emerging Technologies 121 (2020) 102834. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X20307385. doi:10.1016/j.trc.2020.102834.

[16] S. Dabiri, K. Heaslip, Inferring transportation modes from GPS trajectories using a convolutional neural network, Transportation Research Part C: Emerging Technologies 86 (2018) 360–371. URL: https://doi.org/10.1016/j.trc.2017.11.021. doi:10.1016/j.trc.2017.11.021.

[17] M. Cornacchia, K. Ozcan, Y. Zheng, S. Velipasalar, A survey on activity detection and classification using wearable sensors, Ieee Sensors Journal 17 (2017) 7742959. doi:10.1109/JSEN.2016.2628346.

[18] A. N. Koushik, M. Manoj, N. Nezamuddin, Machine learning applications in activity-travel behaviour research: a review, Transport Reviews 0 (2020) 1–24. URL: https://doi.org/10.1080/

01441647.2019.1704307. doi:10.1080/01441647.2019.1704307. arXiv:https://doi.org/10.1080/01441647.2019.1704307.

[19] I. Semanjski, S. Gautama, R. Ahas, F. Witlox, Spatial context mining approach for transport mode recognition from mobile sensed big data, Computers, Environment and Urban Systems 66 (2017) 38–52. doi:10.1016/j.compenvurbsys.2017.07.004.

[20] L. Zhang, S. Dalyot, D. Eggert, M. Sester, Multi-stage approach to travel-mode segmentation and classification of gps traces, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences:[Geospatial Data Infrastructure: From Data Acquisition And Updating To Smarter Services] 38-4 (2011), Nr. W25 38 (2011) 87–93.

[21] A. Kotanen, M. Hännikäinen, H. Leppäkoski, T. D. Hämäläinen, Experiments on local positioning with bluetooth, Proceedings Itcc 2003, International Conference on Information Technology: Computers and Communications (2003) 1197544. doi:10.1109/ITCC.2003.1197544.

[22] F. Subhan, H. Hasbullah, A. Rozyyev, S. T. Bakhsh, Indoor positioning in bluetooth networks using fingerprinting and lateration approach, 2011 International Conference on Information Science and Applications, Icisa 2011 (2011) 5772436. doi:10.1109/ICISA.2011.5772436.

[23] L. Chen, H. Kuusniemi, Y. Chen, J. Liu, L. Pei, L. Ruotsalainen, R. Chen, Constraint kalman filter for indoor bluetooth localization, 2015 23rd European Signal Processing Conference, Eusipco 2015 (2015) 7362717. doi:10.1109/EUSIPCO.2015.7362717.

[24] F. Subhan, H. Hasbullah, K. Ashraf, Kalman filter-based hybrid indoor position estimation technique in bluetooth networks, International Journal of Navigation and Observation 2013 (2013) 570964. doi:10.1155/2013/570964.

[25] H. J. Pérez Iglesias, V. Barral, C. J. Escudero, Indoor person localization system through rssi bluetooth fingerprinting, 2012 19th International Conference on Systems, Signals and Image Processing, Iwssip 2012 (2012) 6208163.

[26] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, E. Aboutanios, Recent advances in indoor localization: A survey on theoretical approaches and applications, Ieee Communications Surveys and Tutorials 19 (2017) 7762095. doi:10.1109/COMST.2016.2632427.

[27] N. Schuessler, K. W. Axhausen, Processing raw data from global positioning systems without additional information, Transportation Research Record 2105 (2009) 28–36. URL: https://doi.org/10.3141/2105-04. doi:10.3141/2105-04.

[28] N. C. Petersen, A. Parslov, F. Rodrigues, Short-term bus travel time prediction for transfer synchronization with intelligent uncertainty handling, arXiv preprint arXiv:2104.06819 (2021).

[29] R. Zhang, P. Isola, A. A. Efros, Split-brain autoencoders: Unsupervised learning by cross-channel prediction, 2016. arXiv:1611.09842.

[30] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[31] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7482–7491.

[32] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, A. J. Smola, A kernel method for the two-sample problem, 2008. arXiv:0805.2368.

[33] I. Tolstikhin, O. Bousquet, S. Gelly, B. Schoelkopf, Wasserstein autoencoders, arXiv preprint arXiv:1711.01558 (2017).

[34] C. Villani, Topics in optimal transportation, 58, American Mathematical Soc., 2003.

[35] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[36] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, arXiv preprint arXiv:1511.05644 (2015).

[37] B. K. Sriperumbudur, K. Fukumizu, G. R. Lanckriet, Universality, characteristic kernels and rkhs embedding of measures., Journal of Machine Learning Research 12 (2011).

[38] A. Likas, N. Vlassis, J. J. Verbeek, The global k-means clustering algorithm, Pattern Recognition 36 (2003) 451–461. URL: https://www.sciencedirect.com/science/article/pii/S0031320302000602. doi:https://doi.org/10.1016/S0031-3203(02)00060-2, biometrics.

[39] K. Khan, S. U. Rehman, K. Aziz, S. Fong, S. Sarasvady, Dbscan: Past, present and future, in: The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), 2014, pp. 232–238. doi:10.1109/ICADIWT.2014.6814687.

[40] M. S. Osman, A. M. Abu-Mahfouz, P. R. Page, A survey on data imputation techniques: Water distribution system as a use case, IEEE Access 6 (2018) 63279–63291. doi:10.1109/ACCESS.2018.2877269.

[41] I. M. Almomani, N. Y. Alkhalil, E. M. Ahmad, R. M. Jodeh, Ubiquitous gps vehicle tracking and management system, in: 2011 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2011, pp. 1–6. doi:10.1109/AEECT.2011.6132526.

[42] 3rd Generation Partnership Project (3GPP), Study on traffic characteristics and performance requirements for AI/ML model transfer, 22.874, 2021. URL: https://portal.3gpp.org.

6 Conclusions

This Ph.D. thesis focuses on a mature field of research, where pushing frontiers proved to be very difficult. Our contribution towards a higher-resolution detector of user travel behavior variations, and a lower bias of travel behavior measures, consists of the following two parts. In the first part we review existing solutions to measure user transport behavior variations, and compare datasets, features, and methodologies across the available literature. This part identifies interesting trends and knowledge gaps (see Chapter 2), which we exploit to solve the problem stated in the introduction (see Chapter 1). In the second part we propose methods for data fusion between GPS and other data sources, and novel artificial neural network architectures to process the data structures resulting from the data fusion, e.g., in the space domain (see Chapter 3). We also expose threats and opportunities of person-to-device and device-to-device interactions for ground truth collection from smartphone devices (see Chapter 4). In the training process of these neural networks, one solution allows for the substitution of human collectible labels with a redundant low cost sensor independent from GPS (see Chapter 5). Finally, as a byproduct of the work described above, we propose (i) signal noise correlation to classification performance and (ii) sensitivity analysis of classification performance to signal noise as two ways to consider the impact of sensor- and label-noise on methods' performance, and thus improve the comparability of different methodologies.

6.1 Trends

Chapter 2 is a review on smartphone-based travel surveys, which have the sole purpose of measuring users' travel behavior variations. The literature contributing to a higher resolution detector of these variations is vast and rests on three main pillars: transport mode detection, trip purpose classification, and map-matching. These three problems appear to be disjoint, and however, we show they have a lot in common. All three could be formulated as classification tasks (perhaps map-matching with some discretizations on the road network), and all rely on GPS as the primary sensor.

To solve the first problem, mode detection, other data often play an essential role, such as INS data, and no data from Geographic Information Systems (GIS) is required. However, GIS data contribution is already positive with simple spatial representations, such as dummy variables.

GIS is necessary for solving purpose imputation tasks, while Person-specific information seems not used for map-matching. However, personal information shows a substantial and positive contribution to the overall models' performance for mode and purpose classification.

Smartphone-based travel surveys may benefit from convergence on the underlying GPS-, GIS-, INS-, and Personal-information. After all, also the underlying GPS segmentation techniques are the same.

We can identify two other mega-convergence trends from the literature: (i) application of method-chains, such as the output of mode detection classification tasks as input for a purpose imputation task, or map-matching output as input for mode detection task; (ii) Bayesian machine learning models implemented with various configurations of artificial neural networks. These two trends seem essential to support any higher-resolution detector of users' travel behavior. From this standpoint, we consider the case of Bayesian methods trained by optimizing Evidence Lower Bound (ELBO), bordering with the artificial neural network framework.

Research and industry are converging towards configurations of smartphones federated learning, leveraging neural networks to be split between smartphones and servers at the edge of the wireless communication network (3rd Generation Partnership Project (3GPP), 2021). The expected data scale could quickly become unmanageable if data processing cannot start immediately after the data generation. Therefore, neural network technology can be considered an essential enabler, facilitating convergence between data structures, modelling methodologies, and processing/compression techniques necessary to support this higher-resolution detector at scale.

6.2 Knowledge Gaps

Datasets in this field are naturally unbalanced. One travels only a tiny fraction of the day, with multiple modes. While not traveling, most of the time, one sleeps, works, and stays at home. However, in the time left, one performs activities of any kind, which he or she repeats daily, weekly, or even yearly. The process leading to a higher-resolution detector for human behavior should also consider the standardization of datasets and labels. Instead, the principal smartphone-based travel surveys agree only on a few of the categories identified within transport mode and trip purposes. Different interpretations on other categories may confuse respondents–for example, exercise can be a trip by bike or jogging, as well as permanence at the gym. In this example, transportation mode represents the first and the second class; trip purpose, the third. Therefore, different survey implementations may introduce different bias on the ground truth, which in turn would confuse the algorithms during the training phase, and contribute to a reduced comparability across studies.

Datasets' standardization requires attention on how to provide realistic representativity and sufficient complexity for the classification tasks one targets. Next, our attention should go on how to harmonize and specify labels to minimize people confusion during validation. Even in the assumption of standardized datasets and labels, the measures of

performance currently in use, as accuracy or f-scores, are just the starting point of a long journey ahead. Computation time, resilience to, e.g., noise levels in the data, or data gaps, could enable case-specific evaluations. Performance drivers should also look at the footprint of the models. For example, in the case of artificial neural networks, we could consider the number of parameters or the memory footprint. Next, we could pinpoint how these models support parameters splitting between smartphone devices and the edge of the communication network, while people is in mobility (3rd Generation Partnership Project (3GPP), 2021). Therefore, the problem of assessing methods' performance looks pretty critical.

Also in this field, datasets and standardization still require continuous work. The lack of both could be attributed to two factors.

1. The person-to-device interaction model currently used for the ground truth collection seems to introduce significant bias in the system. The literature assumes good ground truth, but experiments and datasets' analysis suggest otherwise. In the next section, we will add further arguments on the ground truth. However, instead of labels, Chapter 5 provides experimental evidence that alternative and independent sensors could be exploited for self-validation, and we propose a successful novel architecture for this goal.

2. The literature presents cases where methodological simplifications expose the work to violations of the out-of-sample principle. In some cases, the metric chosen to present the performance seems unable to support the task complexity and the class imbalance (e.g., Accuracy or F1-score weighted average). Emerging transportation modes such as shared mobility–cars, (e)scooters, (e)bikes–put further pressure and compromise sanity checks on transport mode chain types: e.g., one could drive a car to work and use a combination bus and bike to get back home. Lastly, to support such a higher-resolution detector of user travel behavior from smartphones, a dataset representative of all the measures involved in transport behavior requires a magnitude unavailable in the public datasets we found.

6.3 Data Fusion and Machine Learning Models

We focus on two fundamental binary classification problems, both based on GPS signals. The first targets the trajectory segmentation problem to identify stop and motion segments (see Chapter 3). The second targets the identification of users inside or outside the transport network (Be-In/Be-Out - BIBO) (Chapter 4 and 5). The Center of Transport Analytics (DTU) provided the dataset in the first case. Data and ground truth collection relied on a smartphone-based travel survey: the Mobile Market Monitor software. For the second case, we developed and deployed a smartphone sensing platform collecting

GPS from autonomous buses operating a temporary service[1] at DTU campus and their passengers' smartphones, together with the signal strength from a Bluetooth low energy device network installed in the same environment. We collected ground truth in multiple ways, including high-quality from cameras (see Chapter 4) and from personnel following instructions (Chapter 5), as well as lower-quality from users (see Chapter 4).

Chapter 3 proposes a novel technique to fuse GPS signal with the geo-spatial context information collected from GIS. The resulting tensor is defined as a multidimensional high-resolution dummy variable covering each location and its surroundings, consistently with the GPS standard error distribution. We compare the proposed model with other architectures and methods that handle alternative data structures and traditional features–i.e., distance and bearing between GPS points, time of day, and traditional dummy variables representing the surrounding space. The study shows that artificial neural networks relying on the proposed data structure beat Random Forest and advanced clustering methods, and are notably better at handling GPS signal noise. In this work the network architecture is supervised, and the dataset representativeness is sufficient for the binary task at hand. The extension towards a multitask classification of transport mode and purpose would require to fine tune only the last layer of the proposed neural network. However, with the same dataset, branching out the motion class into transport modes and the stop class into activities is impracticable, because the classes are insufficiently represented and unbalanced beyond the limit any model can handle.

Chapter 4 and 5, focusing on Be-In/Be-Out classification, rely on two different datasets. We collect both datasets with the same structure, but with two substantially different scales. Both chapters focus on the use of the BLE signal in the combination with GPS. Chapter 4 explores these signals separately and compares simple classifiers, including native iOS and Android classifiers based on the smartphone's Inertial Navigation System's sensors (INS)–i.e., accelerometer and gyroscope. In our case, where users move and vehicles operate as in high-density and low-speed conditions, these off-the-shelf classifiers based on INS perform similarly to the random classifier. In contrast, simple classifiers based on GPS or BLE signal, such as Random Forest, perform significantly better: Even in presence of flipping labels, also known as labels noise, these combinations signal/classifier seem robust (see Section 6.4 for further details). Chapter 5 explores further both BLE and GPS, proposing a novel cause-effect learning architecture. To focus on the cause-effect relationship and avoid the influence of correlations between BLE and GPS, this architecture extends (i) multitask, (ii) split-brain, and (iii) Wasserstein Autoencoder frameworks. Intuitively, this architecture can use BLE signals instead of labels and achieve effective low dimensional representations, such that DBSCAN algorithms can process it and find clusters consistent with the classification task at hand. This configuration features scalability and flexibility potential beyond any supervised classifier we tested, such

[1] LINC is a large project on self-driving shuttles in Denmark. Further details are available at https://lincproject.dk, accessed in November 2021.

as XGBoost and Random Forest. Even in the assumption of ground truth collectible from person-to-device interactions at high quality, large scale, and low cost, the performance seems comparable. Moreover, the model has a relatively negligible number of parameters, and requires no feature extraction nor data imputation.

6.4 Measures of Ground Truth Collection Errors, GPS Errors, and Impact on Machine Learning

Chapter 3 and 5 focus both on GPS trajectories collected in high-density traffic and slow-speed, typical of urban areas. One chapter looks at how to leverage different signals on the space domain; the other, on the time domain. Both the Chapters cast light on two complementary perspectives about GPS signal and ground truth errors. We attract the attention towards two key performance indexes. The first is the correlation coefficient between GPS noise and classification performance; the second, the classification performance sensitivity to various error rates affecting labels. These two perspectives seem very informative to assess the overall performance of a classifier, including ours. Both give an idea of models' resilience to signal noise, which can derive from multiple factors–one on the sensors signal, the other on the ground truth.

These two perspectives of performance represent an essential contribution for any future standardization process regarding the comparability of competing models. However, the second, regarding noise on labels, seems much more critical. The problem is related to the models' sensitivity to labels' noise, partially solved with semi-supervised architectures that reduce the need for labels. We show that both models and signals are robust to some extension of labels' noise. The biggest challenge is the quality of the ground truth. Data suggests that person-to-device interaction is likely to include noise in the labels collected as ground truth. We highlight that without a controlled ground truth of known quality level, the evaluation of any performance could be subject to very dangerous bias. Whenever applicable, device-to-device or sensor-to-sensor validation, instead of person-to-device, allows for more consistency.

6.5 Contributions and Impact

Within this work, Chapter 2 provides a multi-angle perspective exposing: (i) ongoing methodological and technological convergence in this field; (ii) principal drivers enabling an intuitive comparison of tools and datasets available; and (iii) risks deriving from a blind interpretation of unstandardized measures of performance. Any stakeholder engaging in improving the transportation system can benefit from this basic contribution. Regardless of the technical level of any relevant stakeholder, we provide a tool to improve the critical assessment of solutions aiming at a better understanding of people transport behavior.

Next, Chapter 4 provides a perspective on how ground truth quality impacts transport behavior variation measures based on machine learning. This contribution can be considered as a tool empowering stakeholders, but also as a strong motivation towards a standardized and harmonized process for labels and performance index definition. As in other fields, the ability to reduce the uncertainties on the measures–in this case of behavior variations–would release immense potential.

Lastly, Chapter 3 and 5 dive into powerful data fusion and classification methodologies around GPS signal and artificial neural networks. These contributions enhance the GPS signal exploiting complementary signals on the geo-spatial and temporal domains, leveraging open source Geographic Information Systems and low-cost proximity sensors. On the short term, as results show, these contributions enable more accurate measures of transport behavior variations and lower bias for (i) stop classification, and (ii) presence detection inside transportation systems potentially subject to service disruptions. The first contribution consists of a novel methodology to describe with high resolution the geo-spatial context where trips and activities take place. The second contribution provides a methodology to exploit cause/effect relationship–instead of correlations–for auto-validation of independent signals describing the same event, instead of people validated ground truth. Unlike most the existing methods, both these contributions can handle data and ground truth at the Big data scale. The impact of these methodologies can immediately improve the transport behavior understanding. Below, in a future perspective, we provide further details on their potential for people.

6.6 Future Research

Analyzing users' transport behavior variations in real life exposes users' privacy critically. Any operator is likely to collect sensors streams from users' smartphones, e.g., Geographic Positioning System (GPS) or accelerometer. Thinking of Mobility as a Service (MaaS), the smartphone represents the unified gate to access any transport service. One would expose and collect data within the same domain: (i) Any transport-chain-type, both public and private; (ii) Any personal link-able information related to billing; (iii) Between any origin and destination; and (iv) At any time of the day and activity pattern.

Neither the problem of privacy nor these specific scenarios are new. Smartphone-based travel surveys (SBTS) have the sole purpose of supporting travel patterns discovery and rely on smartphone sensors. The research community in the field is well aware of the problem, represented by the conflict between high data resolution sufficient to study users' travel patterns versus the low resolution necessary to protect their privacy. Solutions proposed since the '90s, seem not able to fulfill SBTS requirements. GDPR (GDPR, 2016) dispensations for SBTS research operators may explain only in part the current limitations. Users' privacy exposure, in this case, relates to user concerns and underpins both recruitment and drop-out problems in SBTS.

Future research should extend the contribution of Chapter 3 and 5 to produce the theoretical advances necessary to: (i) Capture and compress multi-dimension tensorial representations of mobility; (ii) Transfer these compressed representations in a new space where the distributions describing mobility pattern are preserved; and (iii) Ensure that in this new space, the probability of identifying any user approaches zero.

The main gaps that previous studies exposed and that basic research should cover, are the following. Suzuki et al., 2010 showed that the GPS uncertainty necessary to reduce privacy violations is in the range of >100 m from the true position, which seems extremely large compared to the <50 m required for accurate GPS trajectory classification (Zhao et al., 2015). Further, as a future work perspective, Luisa Damiani et al., 2015, and Monreale et al., 2011 claim the need of advancing privacy to, e.g., the geo-spatial and temporal context, which are precisely the domains leveraged for improving, e.g., the classification performance in semantic trajectory generations relevant for mining user transport behavior. Both the contributions provided in Chapter 3 and 5 will be combined to allow accurate behavior detection while preserving privacy, thus with GPS uncertainty in the range of >100 m from the true position.

Another research direction should consider modelling and architectures providing sensor-to-sensor validation by design. This thesis only scratches the surface of the possibilities deriving from the auto-validation of models performing, e.g., classification tasks, while keeping multiple signals describing the same event independently. For example, GPS versus accelerometer, or gyroscope, could show other valuable properties for reducing dependency on labels. This dependency represents a significant bottleneck for any higher-resolution detector of user behavior variations at any larger scale. Therefore, the contribution provided in Chapter 5 will be extended to other smartphones' on board sensors and data structures, including the one presented in Chapter 3.

The data structure of Chapter 3 could also be extended to study the impact of, e.g., altitude taken from topographic maps, and contribute to improve the challenging discrimination between, for example, bikes, e-bikes, and e-scooters.

References

3rd Generation Partnership Project (3GPP) (2021). *Study on traffic characteristics and performance requirements for AI/ML model transfer, 22.874*. URL: https://portal.3gpp.org.

GDPR (2016). "Regulation EU 2016/679 of the European Parliament and of the Council of 27 April 2016". In: *Official Journal of the European Union. Available at: http://ec. europa. eu/justice/data-protection/reform/files/regulation_oj_en. pdf (accessed 20 September 2017)*.

Luisa Damiani, M., E. Bertino, and C. Silvestri (2015). "The PROBE framework for the personalized cloaking of private locations". eng. In.

Monreale, A., R. Trasarti, D. Pedreschi, C. Renso, and V. Bogorny (2011). "C-safety: a framework for the anonymization of semantic trajectories." In: *Trans. Data Priv.* 4.2, pp. 73–101.

Suzuki, A., M. Iwata, Y. Arase, T. Hara, X. Xie, and S. Nishio (2010). "A User Location Anonymization Method for Location Based Services in a Real Environment". In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '10. San Jose, California: Association for Computing Machinery, pp. 398–401. DOI: 10.1145/1869790.1869846. URL: https://doi.org/10.1145/1869790.1869846.

Zhao, F., A. Ghorpade, F. C. Pereira, C. Zegras, and M. Ben-Akiva (2015). "Stop detection in smartphone-based travel surveys". In: *Transportation Research Procedia* 11.2010, pp. 218–226. DOI: 10.1016/j.trpro.2015.12.019. URL: http://dx.doi.org/10.1016/j.trpro.2015.12.019.

This Ph.D. thesis contributes to enabling high-resolution measures of human transport behavior variations from smartphones. Smartphones can contribute to yielding the most prosperous perspective on the study of transport behavior variations both between and within users. While traditional approaches are already measuring behavior variations between users, we need higher resolution to measure these variations within the same user. However, handling such a higher resolution provides a new complex set of challenges.

We pinpoint and examine the problems limiting prior research up-front, exposing drivers to intuitively rank relevant machine-learning algorithms, identify physical limitations, and cast a relationship among human/system interactions, methods, and data. Next, we focus on two fundamental binary classification problems centered on Geographic Positioning System (GPS) trajectories. Both underpin many current and upcoming smartphone-based technologies deployed to measure human transport behavior variations: one problem is "stop" classification; the other is presence detection inside the transport network.

The solution combines GPS time series fused with spatial context information for the first problem. For the second problem, the solution exploits GPS and Bluetooth Low Energy technology. Both solutions rely on the extension of several artificial neural network frameworks based on the back-propagation algorithm. We also study the sensitivity of these methodologies to noise in both sensor signal and ground truth quality. This work underpins novel solutions reducing the dependency on labels and improving comparability across methods.

DTU Management
Department of Technology, Management and Economics
Technical University of Denmark

Akademivej
Building 358
DK-2800 Kongens Lyngby
Tel. +45 45 25 48 00

www.man.dtu.dk