Philipp Grubitzsch, Elias Werner, Daniel Matusek, Viktor Stojanov, Markus Hähnel

AI-Based Transport Mode Recognition for Transportation Planning Utilizing Smartphone Sensor Data From Crowdsensing Campaigns

Erstveröffentlichung in / First published in:

International Conference on Intelligent Transportation. Indianapolis, 19.-22. September 2021. IEEE. S. 1306–1313. ISBN 978-1-7281-9142-3.

DOI: https://doi.org/10.1109/ITSC48978.2021.9564502

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

AI-Based Transport Mode Recognition for Transportation Planning Utilizing Smartphone Sensor Data From Crowdsensing Campaigns

Philipp Grubitzsch[1], Elias Werner[2], Daniel Matusek[1], Viktor Stojanov[1] and Markus Hähnel[1]

***Abstract*— Utilizing smartphone sensor data from crowdsensing (CS) campaigns for transportation planning (TP) requires highly reliable transport mode recognition. To address this, we present our RNN-based AI model *MovDeep*, which works on GPS, accelerometer, magnetometer and gyroscope data. It was trained on 92 hours of labeled data. MovDeep predicts six transportation modes (TM) on one second time windows. A novel postprocessing further improves the prediction results. We present a validation methodology (VM), which simulates unknown context, to get a more realistic estimation of the real-world performance (RWP). We explain why existing work shows overestimated prediction qualities, when they would be used on CS data and why their results are not comparable with each other. With the introduced VM, MovDeep still achieves $99.3\,\%$ F_1-Score on six TM. We confirm the very good RWP for our model on unknown context with the Sussex-Huawei Locomotion data set. For future model comparison, both publicly available data sets can be used with our VM. In the end, we compare MovDeep to a deterministic approach as a baseline for an average performing model (82 – $88\,\%$ RWP Recall) on a CS data set of $540\,\mathrm{k}$ tracks, to show the significant negative impact of even small prediction errors on TP.**

I. INTRODUCTION

Adequate transportation planning (TP) requires information about the movement behavior with Transport Modes (TM) like pedestrians, bicycles, cars, buses, streetcars, trains and their combined usage a.k.a multi-modal transportation in a large urban area. From the movement behavior of a nominal part of the overall traffic, detailed insights about the road network can be derived (e.g. traffic volumes). Crowdsensing (CS), utilizing smartphone sensors to track participants movements is potentially able to provide this information continuously from a huge number of road users without establishing a cost-intense capturing infrastructure in the whole road network.

One major issue with CS are participants who do not track their expected TM. In our CS data set ($3.4\,\mathrm{M}$ tracks) of a city cycling campaign, we observed tens of thousands of non-bicycle tracks recorded by cars, streetcars, buses, trains, and even airplanes or ferries. The impact of single falsely classified trips on the result data for bicycle TP scenarios can be enormous. E.g., in our data set we found a $>600\,\mathrm{km}$ bus trip of identical $10\,\mathrm{km}$ round trips in a small city. Were it recognized as bicycle, traffic volume and average speed information for bicycle TP, would be massively biased. Because bad classification leads to wrong decisions and waste of tax money, a Transport Mode Recognition (TMR) for TP, which works on CS data must be highly reliable.

While traffic engineers use deterministic approaches [1] with limited accuracy, the state of the art for TMR in computer science nowadays bases on artificial intelligence (AI) models and achieves seemingly good results. When studying existing work as a starting point for a concept to process data from our cycling campaign (GPS@$1\,\mathrm{Hz}$, accelerometer (ACC), gyroscope (ROT) and magnetometer (DIR): 3-axes@$100\,\mathrm{Hz}$), we recognized none of them leveraging the full information potential in our data. Moreover, they all neither show any Real World Performance (RWP) nor use a comparable model validation regarding the processing of CS data. Thus, the contribution of our paper is as follows: First, we discuss the conceptual and RWP limits of related work when processing CS data for TP scenarios. In the main part we introduce our AI model concept *MovDeep* to overcome the issues of existing work in detail, including sensor data preprocessing, a new Recurrent Neural Network (RNN) architecture to classify six common TM (feet, bike, car, bus, streetcar, train) on one second time windows and a Post Processing (PP) to further improve prediction results. Moreover, we propose a new validation methodology (VM), which works on unseen data only, to provide a better estimation about RWP of AI models when processing CS data. In the evaluation, we first introduce our data sets and verify our proposed VM. We evaluate the importance of the data from each involved sensor and the chosen scaler. Consequently we show MovDeep with PP achieves $99.3\,\%$ F_1-Score on six TM on our own unseen data set and we confirm the good RWP with the Sussex-Huawei Locomotion (SHL) data set. Both data sets are available publicly for future model comparisons regarding RWP. Finally, we show the negative impact of a model with a RWP of $84\,\%$ F_1-Score compared to MovDeep on a $540\,\mathrm{k}$ tracks CS data set.

II. RELATED WORK

Table I presents related work with focus on AI-based TMR which are considerable for TP. Using location data is subject to signal loss in certain environments [15], e.g. in tunnels or areas of higher population density. ACC and ROT are lacking context for window-based approaches because linear movement with a static device would not be detected, i.e. starting a track inside a train would recognize the smartphone holder as standing still. We examined that each of the sensor provides important information about the TM. Each TM differs in its magnetic footprint, i.e. trains, streetcars and buses have high magnetic anomalies which can be distinctive.

[1]Institute of Systems Architecture, Chair of Computer Networks, TU Dresden [philipp.grubitzsch, daniel.matusek, viktor.stojanov, markus.haehnel1]@tu-dresden.de

[2]Center for Information Services and High Performance Computing (ZIH), TU Dresden elias.werner@tu-dresden.de

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

TABLE I

OVERVIEW OF RESEARCH WORKS RELATED TO TRANSPORT MODE RECOGNITION USING MACHINE LEARNING ALGORITHMS.

Paper	GPS	ACC	ROT	DIR	BAR	TML	MLP	CNN	RNN	PP	TMs	AC(%)	PR(%)	RC(%)	F1(%)	Window Size
[2]	✓						✓				Fe, Bu, Ca	91				tripwise
[3]	✓							✓			Fe, Bi, Bu, Tn	84	86	82	83	200 GPS pts.
[4]	✓							✓			Fe, Bi, Ca	91	90	91	91	tripwise
[5]		✓	✓	✓			✓				Ca, Bu, Sc, Su	93				17,06 s
[6]		✓					✓				Fe, Bu, Ca, Sc	74				n.a.
[7]	✓	✓				✓	✓		✓		Fe, Bi, Ca, Su	93				30 s
[8]		✓	✓	✓					✓		St, Fe, Ru, Bi, Ca			94	94	12 s
[9]		✓	✓	✓	✓				✓		Bu, Ca, Su, Tn	96	96	97	96	1.28 s
[10]		✓	✓	✓					✓		St, Fe, Ru, Bi, Ca, Bu, Tn, Su	96	97	96	96	12 s
[11]		✓	✓	✓		✓				✓	St, Fe, Ca, Bu, Sc, Tn, Su, Fe	95				60 s
[12]		✓	✓	✓	✓			✓	✓		St, Fe, Ru, Bi, Ca, Bu, Tn, Su	98				min/hrs (unclear)
[13]		✓	✓	✓	✓			✓			St, Fe, Ru, Bi, Ca, Bu, Tn, Su				88	5 s
[14]		✓	✓	✓	✓				✓		St, Fe, Ru, Bi, Ca, Bu, Tn, Su				79	5 s (divided in 20 * 1 s overlapping windows)

Global Positioning System (GPS), accelerometer (ACC), gyroscope (ROT), magnetometer (DIR), Barometer (BAR), Traditional Machine Learning Model (TML), Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Post Processing (PP), Transport Modes (TMs), Stationary (St), Feet (Fe), Run (Ru), Bicycle (Bi), Car (Ca), Bus (Bu), Train (Tn), Streetcar (Sc), Subway (Su), Accuracy (AC), Precision (PR), Recall (RC), F1-Score (F1)

The addition of GPS provides an independent interpretation of the actions taking place and is crucial for our use-case of providing reliable data for traffic engineering in order to create maps for TP. These four sensors are the most common across devices and – when combined – provide detailed information about the tracking users.

Compared to Traditional Machine Learning Models (TMLs) (e.g. Random Forests, Support Vector Machines) and Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs) and RNNs can combine significant amounts of data and take a more informed decision by including context based data into the equation. The latter are able to make use of a temporal dependency between the time windows, which is crucial for a window-based TMR where multi-modal transportation is common, i.e. changes in TM can occur any time. To detect those possibly frequent TM switches, the window size must be as small as possible but sufficiently sized for a reliable classification. If possible, RNNs achieve this by combining weighted data from the past, present and future, when predicting a given slice of measurements.

As shown by [11], the prediction results can be significantly improved by a PP algorithm, which corrects erroneous predictions. Their approach relies on the fact, that switching vehicular TM presupposes a feet intersection and improved their results significantly. Nevertheless, their approach works on $60\,\mathrm{s}$ time windows and thus does not address other prediction errors on shorter time windows.

Because of different available data and requirements, the set of classified TMs vary between most papers. Most of them only include a small set or have been specified on specific types, i.e. public transport [9]. While their methods might work in these specific cases, adding new TMs changes the requirements because of the similarities and the peculiarities the classification algorithm needs to learn.

[9], [11], [10] and [12] are the works with the most promising results (scores $>95\,\%$). Although [9]'s model can only predict four TMs, it is included because of its architecture and high accuracy. They all achieve these results even without using GPS, which retains the possibility to still add GPS to their approaches to meet the requirements of TP. However, as our own research of feature sets shows, the GPS still provides the most significant information for an AI-model. Moreover, all approaches work with time windows $>1\,\mathrm{s}$. [9] is close enough but can only predict four TM. CS data likely contains unknown patterns which appear as outliers for the trained AI-model. All four works use Z-score normalization, which doesn't account for context drift. A robust scaling approach could be Yeo-Johnson transformation [16] which is capable of dealing with outliers. Notably, almost all investigated approaches split their training, validation and test data sets randomly. As we will show in this paper, this does not provide any good estimation about the RWP. Only [9] considered this and conducted first experiments by testing their model with unseen data. After applying their validation approach, the accuracy dropped from $96.9\,\%$ to $92.3\,\%$, but provide a more realistic RWP. Unfortunately, they do not provide a detailed description of how their VM is conducted. We conclude, all of those approaches should also prove their reliability with respect to a RWP by testing them against unknown data. This can be conducted with our publicly provided data set. As our own research will show the information in the GPS, the choosing of the right scaler and independent validation data is crucial for achieving reliable RWP with CS data. Hence, we tested our RNN model against a publicly available subset of the SHL data set and not only our own data set, to get a reliable estimation of the RWP.

To emphasize the unfullfilled requirements for TP of the related work, we assess two promising projects in the field of TMR which participated in the SHL 2020 challenge [17]. Firstly, [13] is the best performing approach regarding the F_1-score. In [13] the authors rely on a CNN approach in which they automatically extract features for an afterwards classification of the transportation mode. Secondly, [14] is best performing with a RNN. [14] are using a similar approach to ours besides a missing PP. They pre-process data to calculate direction-independent vectors of the acceleration and the compass data and manually calculate hand-crafted

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

features for their RNN. Both approaches use training, validation and test data which were collected by only three persons in only the same area, which introduces bias when it comes to predicting unseen data. As already noticed in our experiments and by [9], introducing completely unseen data from new users, new locations and other devices drastically decreases the accuracy and prediction results. Although the approaches by [13] and [14] show great results on the SHL data set, we cannot make assumptions about their performance when predicting this completely unknown data. Additionally, in order to provide reliable data for TP with a CS data source, a F_1-score of $79\,\%$ and $88\,\%$ respectively still remains space for improvement. Another reason why their concepts do not suit our use case is the usage of the barometer instead of the Global Positioning System (GPS) signal. Our as well as other CS campaigns do not provide data about the air pressure, thus we cannot use the models trained by [13] and [14]. Nevertheless, this promising approach could be considered in the future with an extension on the CS data provider's side.

Concluding, none of the examined works fully meet our requirements for a window-based approach on short time windows with context- and time-dependent predictions using all available sensor data. They are either missing a suitable postprocessing of the predicted data, or can only predict a small set of TMs and do not provide a realistic estimation of the RWP, which is crucial for the application on CS data. Our work aims at filling this gap.

III. CONCEPT

In this section, we present our overall concept. At first, we explain the preprocessing, that extracts features from the raw sensor data. Then, we introduce our RNN architecture and describe important hyper parameters. Subsequently, our improved postprocessing is explained. Finally, we propose a new VM for better RWP estimations.

A. Preprocessing

We utilize the raw data from GPS, ACC, DIR and ROT over time t. For GPS, we obtain $\vec{r}(t_k) = (r_{\text{lat}}, r_{\text{lon}})^\top$ with latitude r_{lat} as well as longitude r_{lon}, and the speed $v(t_k)$ at a sampling rate of $1\,\text{Hz}$. The ACC offers $\vec{a}(t_l)$ for three axes at an average sampling rate of $100\,\text{Hz}$. The same applies to ROT $\vec{\omega}(t_m)$ and the DIR $\vec{m}(t_n)$.

The related work confirmed that feature extraction is the best approach to provide information to an AI-model for TMR and to achieve the best overall performance. Hence, we extract features from the raw sensor data. In total, we investigated over 3000 heuristics in the time and the frequency domain of 152 preprocessed values for their significance to distinguish between the TMs. Due to space limitation we explain only the most expressive, finally used ones.

In addition to the speed $v(t_k)$, the acceleration is calculated between two sampling points: $a_r(t_k) = v(t_k) / (t_k - t_{k-1})$. Furthermore, we consider the changes in the direction of the movement. We include $\Delta\varphi(t_k) = \measuredangle\left(\vec{r}(t_{k-1}), \vec{r}(t_k)\right)$ in the interval $[-\pi, \pi]$ which distinguishes between left/right as well as its absolute $\left|\Delta\varphi(t_k)\right|$. For the ACC, we consider the norm $a(t_l) = \left|\vec{a}(t_l)\right|$ as well as the norm of increment $\Delta a(t_l) = \left|\vec{a}(t_l) - \vec{a}(t_{l-1})\right|$. The same is being conducted for ROT ($\omega(t_m)$, $\Delta\omega(t_m)$) and DIR ($m(t_n)$, $\Delta m(t_n)$).

The vertical and horizontal portions of the sensors $\vec{a}$, $\vec{\omega}$ and $\vec{m}$ contain important information. Indeed, the direction's frame of reference is the device's orientation, which may change during acquirement due to movements of the mobile device. Instead, it is appropriate to define these sensors' frame of reference as the fixed outer world, namely north, east and altitude. Thus, we do a transformation of coordinates to a altitude heading reference system (AHRS). Inspired by [18], we calculate the adjusted stationary sensors $\vec{a}_{\text{AHRS}}(t_l)$, $\vec{\omega}_{\text{AHRS}}(t_m)$ and $\vec{m}_{\text{AHRS}}(t_n)$.

Next, the different sampling points $t_{\{k,l,m,n\}}$ are synchronized. Therefore, a down-sampling to $1\,\text{Hz}$ is applied in order to achieve common timestamps t_i. All preliminary values in the interval $[t_i, t_{i+1})$ are cumulated by the $\text{mean}(\cdot, t_i)$. At the same time, we obtain the Boolean "stop" as characteristic between different TMs. We use a basic definition based on the standard deviation (std) of the ACC yet: stop at $t_i :\Leftrightarrow \text{mean}(v, t_i) \leq 0.5\,\text{m/s} \wedge \text{std}(a, t_i) \leq 0.5\,\text{m/s}^2$.

Finally, missing values of the three sensors are filled by cubic interpolation $\mathcal{I}(\cdot)$. Since GPS is missed for longer periods of time while being in buildings or underground, these missing values are discarded instead. In summary, our preprocessing approach uses 20 features in a single feature window f_i with the timestamp t_i and a length of one second:

- **5 GPS features**: $\text{mean}(v, t_i)$, $\text{mean}(a_r, t_i)$, stop, $\text{mean}(\Delta\varphi, t_i)$ and $\text{mean}(\left|\Delta\varphi\right|, t_i)$
- **3 sensors** as $\vec{s} \in \{\vec{a}, \vec{\omega}, \vec{m}\}$ with features
  - **2 scalar**: $\mathcal{I}(\text{mean}(s, t_i))$ & $\mathcal{I}(\text{mean}(\Delta s, t_i))$
  - **1 vectorial** (3 components): $\mathcal{I}(\text{mean}(\vec{s}_{\text{AHRS}}, t_i))$

Scaling input data is a proven and widely used technique to improve the prediction quality of an AI-model. Therefore, we scale the features with Yeo-Johnson transformation [16] into a smaller range of values before passing them to the model. The advantage of this scaling technique is its outlier-robustness in contrast to other scaling approaches, such as Min-Max or Z-Score scaling.

B. RNN Architecture

Our TMR approach bases on Recurrent Neural Network (RNN) to consider time-variant data over a larger time span but still preserve the ability to predict TM on short time windows. Short time windows ensure the precise recognition of changes in case of multi-modal transportation on single tracks and thus improve the prediction quality of the model. The network architecture and explored hyper-parameters are depicted in Fig 1. Each window of the windowed feature sets has a size of one second and consist of the 20 features as described in subsection III-A. To predict an TM for the present window p_i, the approach submits the related windowed feature set (FS) f_i to a feed forward sub net. Moreover, it utilizes multiple FS for n windows in the past

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

and m windows in the future by submitting them to distinct recurrent sub nets, extracting the right information from the right time. All the information from past, present and future is *concatenated* and provided to a subsequent feed forward sub net that makes the final decision of the present TM. Short time windows contain very limited information to classify the TM. To overcome this, a network should be trained which considers information from past and future. This design enables the approach to detect patterns over multiple time windows by incorporating information from them. For instance, the regular stop of a public transport can be detected and used as an indicator for Bus, Streetcar or Train.

Based on our 20 features, the given width, depth and activation where examined and yield an optimum for the outlined parameters. Regarding the considered time spans of past and future, $n = m = 90$ seconds revealed a prediction sweet spot by preliminary examinations. Furthermore, this limited time span has lower computational cost, which supports our goal to process a huge real world CS data set. Note that the length of n and m depends also on the TM and context to interact with and can vary in different use cases. For instance, a bicycle can be distinguished from a train in a much shorter time window, whereas the distinction of a train and a car needs longer term information. If there is no information available for the past or the future (beginning/end of a track), the approach handles it by padding this information. Moreover, up to real time predictions are enabled by the independence of n and m. This is achieved by considering shorter future time spans or only utilizing data from the past.

C. Post processing

The predicted windows p_i with $1\,\mathrm{s}$ length as output of our RNN, are prone to an unsteady prediction labeling. Thus, we introduce a PP to correct implausible labels, which is inspired by the healing in [11]. They exploit the logical presence of feet segments between different TM, e.g. Bus→Feet→Streetcar. It relabels window sequences of vehicular predictions, that have missing feet intersections, e.g. Bus→Streetcar. Such sequences are then relabeled as one TM by majority vote. Nevertheless, their approach relies on a precise recognition of Feet and only works reliable on time windows $\geq 60\,\mathrm{s}$. Thus, we extend their concept by two additional steps. Fig 2 gives an example for an unsteady prediction labeling p_i and how the labels are relabeled by our extended PP to get the corrected labels p_i'. An introduced first step mainly addresses unsteady predictions on stop window sequences between all TM. It relabels windows as the previous TM, if the preprocessing calculated them as a stop, e.g. recognition of Car during a bus stop. The second step represents the original approach by [11]. Unsteady sequences calculated as non-stop windows are a similar open issue, e.g. a walking person in a bus at a bus stop. Hence, window-based smoothing as a third step is introduced. It checks the label of the present window and considers the predicted labels $30\,\mathrm{s}$ from the past and $30\,\mathrm{s}$ to the future. Within their time span, the major labels for the past and the future are independently determined by majority vote. The presence label is then checked and not relabeled if it matches either the past or the future major label. Otherwise the presence label is relabeled with the past major label.

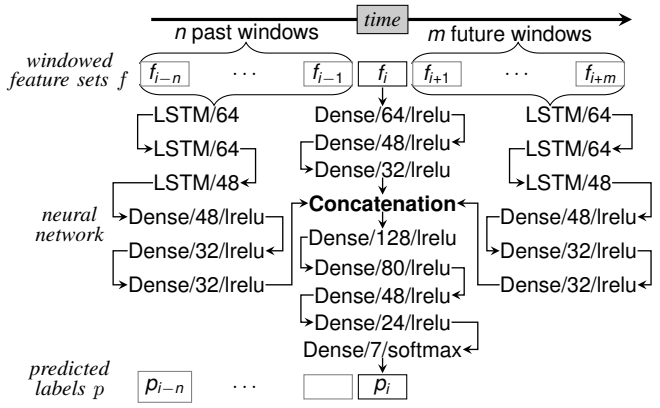

Fig. 1. TMR using a series of windowed FS by applying an RNN architecture

D. Validation Methodology to estimate RWP

As already stated, almost all related work randomly split one data set for training and validation (e.g. 80/20). When examining our concept, we worked with a continuously growing hand-labeled data set, gathered by several persons. We trained our model with the random split at specific state/size of this data set and could achieve very good prediction results (F_1-Score $>95\,\%$). After other persons provided more tracks, we used these tracks to validate the latest model again and noticed heavy drops of up to $10\,\%$ F_1-Score. We assumed, that our model applied on our CS data set, would not perform as well as our initial validation would suggest. That means, the estimated RWP is much worse than expected. [9] firstly discussed the questionable representativeness when doing a random split, but unfortunately provided no detailed description for an improved VM.

Before presenting our approach, we give the detailed reason for the drop in F_1-Score as follows. The underlying data is not only characterized by TM specific patterns, but also by user, device and location specific ones. During the training, the model might have adapted to all these additional patterns of the training data, as well. An randomized 80/20 split implies for a RWP, that these additional patterns would also occur in the same manner. But this never happens

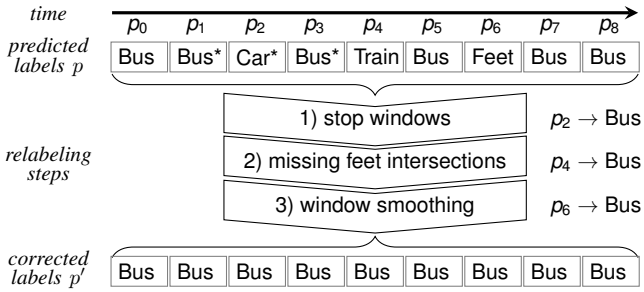

Fig. 2. Brief overview of PP. Note: $*$ is a calculated stop

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

TABLE II
DURATION IN HOURS AND TRAINING/VALIDATION SPLIT OF DIFFERENT TMS OF MOVDATA.

TM	Streetcar	Feet	Bus	Bicycle	Train	Plane	Car	Total
Training	10.13	27.87	7.47	17.41	10.38	1.63	17.19	92.09
Validation	0.72	4.80	1.58	4.67	3.52	0.00	2.43	17.05
Combined	10.84	32.00	9.05	22.08	13.91	1.63	19.62	109.14

completely. Hence, we assume, any model will perform better on such validation data, than it would be in the real world. An appropriate VM to estimate the RWP for TMR should reveal this weaknesses of an AI model, because the model should mainly learn the TM pattern, and ideally ignore all other patterns.

We overcome this issue by presenting a new VM, where training and validation data should be split on independent persons, devices and locations. Then, the validation data, will contain new individual patterns like movement behavior, locations with related anomalies (esp. no GPS signal, magnetic field), values/ranges from other integrated mobile device sensors and even different carrying locations (hand, backpack, pants, bicycle handlebars etc.). Moreover, we recommend to build a validation data set that is as much heterogeneous as possible regarding these points. If this is not considered and the validation data set consists only of one person, device or location, there are only two extrema: 1) The trained AI model matches the validation data well, while this person, device or location might not be representative for all people. Then the obtained RWP estimation would be too good. 2) The AI model matches validation data bad. Then the obtained RWP estimation could also be too bad. In the evaluation part, we verify the advantage of our proposed VM to compare AI models for TMR in general.

IV. EVALUATION

Before we verify our concept, we introduce the evaluation environment and data sets. The first experiment shows the advantage of our VM to estimate the RWP. Afterwards, we estimate the RWP of our RNN model and the positive impact of the proposed scaler and the introduced PP. In the last experiment we apply our model to a $540\,\mathrm{k}$ tracks sized CS data set from a cycling campaign and compare the result to a state-of-the-art approach, in practice applied by transportation engineers. We verify that a high reliability of TMR is crucial for utilizing CS data in the TP domain.

A. Evaluation Environment and data sets

Our evaluation utilizes three data sets. Our own training and validation data set was gathered and labeled manually by six users carrying different smartphones, for six TM. No explicit instructions how to carry devices or to behave while tracking have been declared. Table II summarizes the collected data. Following, we name this data set MovData.

Additionally, we use the public available SHL[1] data set as completely unseen data set. To match our TMs, we map SHL's Walk & Run to Feet, Train & Subway to Train and don't consider SHL's TM Null & Still. Note: there is no Streetcar class anymore, because this class does not exist in SHL. At last, we work with a CS data set provided from three-year cycling campaign, where more than $160\,\mathrm{k}$ people recorded over $3.5\,\mathrm{M}$ tracks, containing $70\,\mathrm{TB}$ raw sensor data of various TM. All data sets contain the required sensor data (GPS,ACC,DIR,ROT) and the six supported TMs.

We implemented the concept in python3, using Keras 2.3.0 with Tensorflow 2.1.0, sklearn 0.22.0, pandas 1.0.3 and numpy 1.18.1. The RNN is trained with early stopping, L2 regularization (factor 0.0001) and Dropout (rate 0.2).

B. Investigating the VM

We conducted a validation of our model in three ways to show the advantage of our proposed VM over a randomzied 80/20 split when estimating RWP. Note that we do not use our PP here, since we want to evaluate what the AI has learned instead of focusing on the potential of the PP. First, a randomized 80/20 split on MovData yields a F_1-Score of $94.23\,\%$. Second, our proposed VM from subsection III-D estimates a F_1-Score of $89.95\,\%$ on the same data set, which shows a first RWP drop for unseen data. Third, we conducted our VM on the completely unseen SHL data set, which results in a F_1-Score drop to $77.99\,\%$. This strengthens the assumptions about the importance of the disjointness of training and validation data sets as outlined in subsection III-D. Moreover, it highlights the need of heterogeneity not only in the training data, but also in the validation data for a representative RWP estimation. Related TMR approaches, trained and evaluated their models on data sets with limited users only (e.g. SHL).

C. Verifying the Need for Various Sensors

To show the information provided by each sensor, we applied our approach for GPS, ACC, ROT and DIR separately. Fig 3 shows the recall and precision of each sensor. We interpret the achieved results as follows:

a) GPS: is a very good basis to recognize Feet and Bicycle as well as Car and Train. On the other hand Streetcar and Bus are hard to distinguish with GPS only. The main patterns in the GPS signal are the speed, the acceleration and the 2-dimensional movement. That makes it difficult to distinguish inner city streetcar from bus and bus even from bike as these patterns can be very similar.

b) ACC: is probably the most widely available sensor in addition to GPS. The best results can be achieved for Feet and Bike. Bus and Train can be reasonably good distinguished. Notably the ACC can heavily improve the recognition of the Bus compared to GPS only. Streetcar and Train are often mixed up, whereas the Car class is heavily mixed with Bus. The main pattern in the ACC signal is shaking, which is the reason for the confusion between Bus and Car. Streetcars have different shaking pattern compared to a Bus, because it goes on rails, which improves their distinctness. On the other hand, a Streetcar has a very similar shaking pattern to Train, which causes their confusion.

[1] http://www.shl-dataset.org/dataset/, [19] and [20]

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

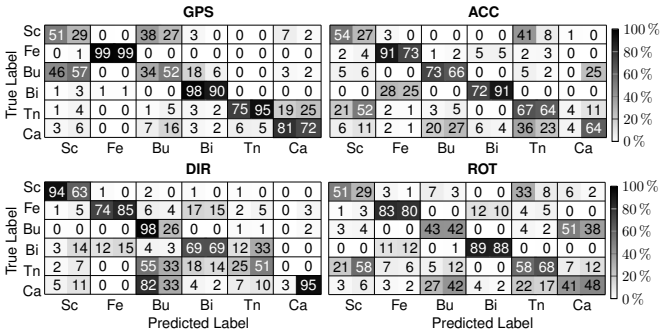

Fig. 3. Prediction of the MovData validation data set after training on the MovData training data set with restriction to specific sensors. Each cell contains the Recall on the left as well as the Precision on the right (in percent). Labels: Streetcar (Sc), Feet (Fe), Bus (Bu), Bicycle (Bi), Train (Tn), Car (Ca)

c) DIR: measures the orientation with respect to the earth's magnetic field. It explicitly acquires the direction of movement and implicitly observes turn patterns. However, on a short time scale they are both superimposed by the magnetic fields of electric motors or massive iron chassis. Actually these magnetic anomalies provide significant information to distinguish Streetcar (overhead wiring and/or electric motor) and Bus (chassis and/or electric motor). With mostly no magnetic field involved also Feet is very good predicted, as pedestrians can be distinguished from other TM by their specific turn patterns.

d) ROT: introduces the benefit of recognizing turn patterns explicitly. For the angular velocity we assume: $Train < Car < Bicycle < Feet$. While trains have a rather huge turning circle, pedestrians can turn on the spot. Hence, ROT is crucial to distinguish Train vs. Car and works good for Bicycle vs. Feet. In contrast to DIR, it is only superimposed by a small drift over longer time by warming. Hence, it measures turns on a short time scale accurately.

e) Discussion of Sensor fusion advantage: As discussed above, no single sensor alone yields a sufficient result for all TM. Every sensor is predestined for one or more important but not unique aspects of the TMs. But, providing a combination of sensor information to the AI model can improve the distinction of problematic classes. For example, with GPS only, Bus is mainly confused with Streetcar, Bike and Car. The ACC and ROT support filtering Streetcar and Bike and finally DIR filters Car. Thus, the combination of all sensors helps to get a much better distinction than only one sensor could.

D. Choosing the Right Scaler

Next, we show the importance of choosing the right scaler regarding the occurrence of outliers. We implemented both the Min-Max scaling and the Yeo-Johnson transformation. While validating with our data set only, we obtain similar F_1-scores of 89.46 and 89.95 %, respectively. On SHL we obtain 71.24 and 76.66 %. Hence, Yeo-Johnson outperforms Min-Max scaling in case of context drift. The need for a reliable scaler becomes important when working with CS data, where context drift is to be expected. Nevertheless, the classes Car and Bus are confused. This can be improved, by adding more samples to the training data set and thus increasing its heterogeneity.

E. Positive Impact of Post Processing regarding RWP

After taking into account the previously discussed steps, our model already achieves 89.95 % F_1-score on our validation data set. Applying PP as described by [11], the model prediction is improved to 97.02 %. Adding our two novel steps (see subsection III-C), further improves F_1-score to 99.30 %. As outlined in Fig 4, this is a nearly perfect result. Misclassifications only appear within the Feet class, what can be explained by an imprecise manual labeling, caused by the 1 Hz data rate. This affects public transport classes, since waiting before departure tends to be inaccurately labeled.

Fig 5 shows the results with PP on SHL. We achieve an overall F_1-Score of 85.22 %. In consideration that the predicted data underlies unseen devices, users and even locations, this result is representative for the RWP of our model can be assessed as very good. This is in sharp contrast to related work, that trained their approach already on the SHL data set. Thus, their models had already seen the underlying patterns and were adapted to them. As explained in subsection III-D, this makes it easy to achieve a good prediction result but does not represent the RWP. The drop of our prediction quality from the MovData validation data set to the SHL data set, can be explained as a result of our training on a data set, that is not heterogeneous enough. Especially, the bus class has a low F_1-Score of 65.75 %, meaning that it is not represented adequately in the training data set. Again, adding data of the same TM but with different user, device and location specific patterns would improve that behavior by adding more heterogeneity to the training data.

F. Verifying Reliability of TMR on CS data sets

1) Towards a Baseline of a TP State of the Art Approach: As mentioned in Sec II, no existing approach based on neural networks fulfills the requirements nor evaluates with disjoint data. Hence, we compare MovDeep with the deterministic solution presented in [1], called MovOrig in the following. This is currently state-of-the-art in the domain of bicycle TP. Note that for the comparison, the prediction labels are limited to the intersection of both approaches, namely

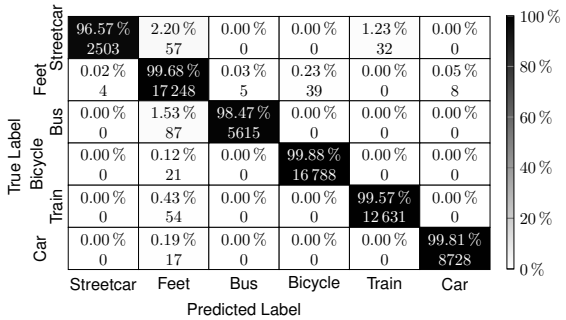

Fig. 4. Prediction of "MovDeep" on the MovData validation data set

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

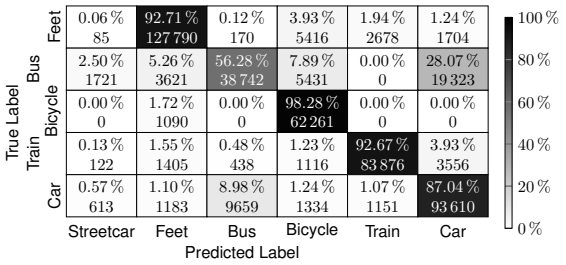

Fig. 5. Prediction of "MovDeep" with PP on the SHL data set

the classes Feet, Bicycle and Other. Moreover MovOrig introduces an Activity filtering, which is supposed to prefilter non-modal track segments, to recognize multi-modal changes in a track. First, we compare MovDeep and MovOrig on our own data set. As summarized in Table III and Fig 6, MovDeep classifies the crucial Bicycle class nearly perfect. In contrast, MovOrig labels feet samples as Bicycle what lowers the precision and does not recognize all bicycle samples correctly what causes a lower recall. The Feet class is also only poorly recognized, due to the activity filtering, that is not able to differentiate between a walk with and without a transport purpose. This leads to an overall F_1-score of $62.20\,\%$ and $83.69\,\%$ for the bicycle class. When predicting on SHL, interestingly the overall F_1-score of MovOrig increases to $69.01\,\%$, mainly because of an improved F_1-score of $56.53\,\%$ for the Feet class. Contrary, the F_1-score for Bicycle decreases to $66.22\,\%$. Therefore, the requirements of bicycle TP are not fulfilled by this TMR due to the under-representation. We conclude, MovDeep significantly outperforms MovOrig with a F_1-score of $93.79\,\%$ ($90.63\,\%$ for Bicycle) regarding an estimated RWP.

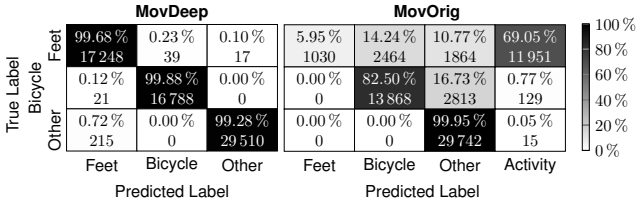

Fig. 6. Comparison of the "MovDeep" and "MovOrig" on the MovData validation data set with class restriction

2) MovDeep vs. MovOrig on 540 k CS data set: To demonstrate the discrepancy between MovDeep and MovOrig, we predicted $2.2\,\mathrm{M}$ tracks from the mentioned cycling campaign in 2020. It contains various TM tracks but also gaps in GPS or sensor data. Since handling missing sensor data relies on the pre-processing only, we filtered tracks that contain gaps, to achieve as trustworthy results as possible. Thus, we finally used $540\,\mathrm{k}$ tracks to compare MovDeep and MovOrig.

TABLE III
PRECISION (PR), RECALL (RC) AND F_1-SCORE (F1) FOR THE CLASSES FEET, BICYCLE AND OTHER BY MOVDEEP AND MOVORIG

Mov	MovDeep			MovOrig		
	PR(%)	RC(%)	**F1(%)**	PR(%)	RC(%)	**F1**
Feet	98.65	99.68	**99.16**	100	5.95	**11.23**
Bicycle	99.76	99.88	**99.82**	84.91	82.50	**83.69**
Other	99.94	99.28	**99.61**	86.41	99.95	**92.69**
SHL	MovDeep			MovOrig		
	PR(%)	RC(%)	**F1(%)**	PR(%)	RC(%)	**F1**
Feet	95.80	92.71	**94.17**	87.86	56.53	**68.79**
Bicycle	84.09	98.28	**90.63**	52.88	88.57	**66.22**
Other	97.78	95.35	**96.55**	95.81	57.66	**71.99**

MovDeep identifies $2.79\,\mathrm{M\,km}$ as Bicycle. Even when assuming the most unlikely worst-case from SHL with the Precision of $84.09\,\%$ and the Recall of $98.28\,\%$ (see Table III), the relative uncertainty is in the order of magnitude of $10\,\%$. Nevertheless, this can be considered as "true" with respect to Fig 6 and, hence, is our reference.

In contrast, MovOrig labeled only $2.59\,\mathrm{M\,km}$ as Bicycle. Hence, we next compared the amount of distance identified as Bicycle for each of the $540\,\mathrm{k}$ tracks. If MovOrig identified less distance, we assume that the precision of the Bicycle class was 1 and the missing distance was falsely identified as Feet, Other or Activity. Contrary, if MovOrig identified more distance as Bicycle, we assume that the recall of the Bicycle class was 1 and the over-estimated distance is within Feet, Other and Activity. Both assumptions are in favor of MovOrig and so represent its best-case. Nevertheless, $261\,993\,\mathrm{km}$ were not identified as Bicycle (false negatives) while $62\,412\,\mathrm{km}$ were incorrectly identified as Bicycle (false positives). This yields a relative uncertainty of $12.5\,\%$. Up to this point, we conclude that even MovDeep's worst-case outperforms MovOrig's best-case.

MovOrig is completely dependent on GPS data and thus has to discard sections without GPS data. However, MovDeep can handle these sections, since it considers ACC, ROT, DIR and does not rely on GPS only. As shown in subsection IV-C, a TMR is possible even with single sensors. Thus, MovDeep would still be applicable if some sensor data is temporary or always not present, whereas MovOrig is not. Additionally, the filtering of the $540\,\mathrm{k}$ tracks for the trustworthy comparison biases TM in favor of cyclists with positive effects for the given results of MovOrig. Especially the Train class was filtered due to a bad GPS signal. We assume, that a more heterogeneous, general data set would lead to a significant decreased performance of MovOrig.

In contrast to [1] who stated, MovOrig provides a sufficient performance, we must conclude this is at least questionable. As we have shown, applying MovOrig on the CS data set introduces a significant bias of recognized TM and therefore resulting in wrong information for TP applications. For example, expecting mainly bicycle tracks, but including tracks from pedestrians and falsely excluding true bicycle tracks results in too low speed information on specific road segments for a speed map.

V. CONCLUSIONS

The overall motivation of our work was to process CS data from a cycling campaign to offer detailed and reliable information for bicycle TP. We summarize our contribution as follows. We discussed, why previous work does not address the related requirements to handle the particularities of non laboratory CS data while still achieve reliable good

Final edited form was published in "International Conference on Intelligent Transportation. Indianapolis 2021", S. 1306–1313, ISBN 978-1-7281-9142-3
https://doi.org/10.1109/ITSC48978.2021.9564502

prediction results. Even more we argue, that the overwhelming majority of approaches is not validated in terms of a realistic RWP estimation on CS data. We state here, that an appropriate VM should be chosen not to highlight the strengths of AI models, but to highlight their weaknesses. Only such an VM can provide a good estimation about the RWP. Thus, we proposed a novel VM, where the split for training and validation data is conducted on independent persons, devices and locations. To process our CS data set, we proposed *MovDeep*, a novel TMR approach in detail, that classifies six TMs, demanded by TP. Our evaluation confirmed that our proposed VM can improve the reliability of the RWP estimation. We evaluated MovDeep with our VM and achieved $99.3\,\%$ on our own and $85.2\,\%$ on the SHL data set. In this context, we verified positive impact of all chosen sensors, of the proposed scaler and the extended PP. Finally, we have shown that an average performing approach in terms of RWP should not be utilized to process CS data, as it would massively biases the TP applications. We also provide the MovData validation data set[2] to the public. Thereby, other authors can use SHL and MovData to estimate the achieved RWP of their model and compare it with our and other works.

Our future work aims for optimising various points of our developed approach. First, we consider the integration of height information from both barometer and GPS data. We expect even more accurate and reliable prediction results with the new gathered information. We want to implement an anomaly detection and handling for erroneous and incomplete input data. Moreover, we want to optimize our PP approach for different TM. At this point, every TM is post-processed using the same smoothing window, which is sub-optimal. Furthermore, we aim for improving our evaluation. On the one hand, we are planning to perform a detailed k-fold cross-validation on the SHL data set and on MovData. Additionally, a random sample analysis will be carried out on the CS data set checking the plausibility of the recognised TM to potentially identify remaining problems. Finally, we plan to publish an extended data set containing more tracks, users and unique devices recorded at different locations.

ACKNOWLEDGEMENTS

The work for this paper has been conducted during the research project *Movebis*[3] (fund number 19F2011B) and is funded by the Federal Ministry of Transport and Digital Infrastructure in Germany. The authors would like to thank the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) for its support to publish this article.

REFERENCES

[1] S. Lißner and S. Huber, "Facing the needs for clean bicycle data–a bicycle-specific approach of gps data processing," *European Transport Research Review*, vol. 13, no. 1, pp. 1–14, 2021.

[2] P. Gonzalez, J. Weinstein, S. Barbeau, M. Labrador, P. Winters, N. Georggi, and R. Perez, "Automating mode detection for travel behaviour analysis by using global positioning systemsenabled mobile phones and neural networks," *IET Intelligent Transport Systems*, vol. 4, no. 1, pp. 37–49, 2010.

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

[4] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Ensemble Convolutional Neural Networks for Mode Inference in Smartphone Travel Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2232–2239, apr 2019.

[5] S.-H. Fang, Y.-X. Fei, Z. Xu, and Y. Tsao, "Learning Transportation Modes From Smartphone Sensors Based on Deep Neural Network," *IEEE Sensors Journal*, vol. 17, no. 18, pp. 6111–6118, Sept. 2017.

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

[7] S. Ferrer and T. Ruiz, "Travel Behavior Characterization Using Raw Accelerometer Data Collected from Smartphones," *Procedia - Social and Behavioral Sciences*, vol. 160, pp. 140–149, Dec. 2014.

[8] T. H. Vu, L. Dung, and J. C. Wang, "Transportation mode detection on mobile devices using recurrent nets," in *MM 2016 - Proceedings of the 2016 ACM Multimedia Conference*. Association for Computing Machinery, Inc, oct 2016, pp. 392–396.

[9] H. Wang, H. Luo, F. Zhao, Y. Qin, Z. Zhao, and Y. Chen, "Detecting transportation modes with low-power-consumption sensors using recurrent neural network," 10 2018, pp. 1098–1105.

[10] G. Asci and M. A. Guvensan, "A novel input set for lstm-based transport mode detection," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2019, pp. 107–112.

[11] A. Guvensan, B. Dusun, B. Can, and H. Turkmen, "A novel segment-based approach for improving classification performance of transport mode detection," *Sensors (Basel, Switzerland)*, vol. 18, 12 2017.

[12] Y. Qin, H. Luo, F. Zhao, C. Wang, J. Wang, and Y. Zhang, "Toward Transportation Mode Recognition Using Deep Convolutional and Long Short-Term Memory Recurrent Neural Networks," *IEEE Access*, vol. 7, pp. 142 353–142 367, 2019.

[13] Y. Zhu, H. Luo, R. Chen, F. Zhao, and L. Su, "Densenetx and gru for the sussex-huawei locomotion-transportation recognition challenge," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 373–377.

[14] B. Zhao, S. Li, and Y. Gao, "Indrnn based long-term temporal recognition in the spatial and frequency domain," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 368–372.

[15] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems - SenSys '13*. Roma, Italy: ACM Press, 2013, pp. 1–14.

[16] I.-K. Yeo and R. Johnson, "A new family of power transformations to improve normality or symmetry," *Biometrika*, vol. 87, 12 2000.

[17] L. Wang, H. Gjoreski, M. Ciliberto, P. Lago, K. Murao, T. Okita, and D. Roggen, "Summary of the sussex-huawei locomotion-transportation recognition challenge 2020," 09 2020.

[18] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[19] H. Gjoreski, M. Ciliberto, L. Wang, F. J. Ordonez Morales, S. Mekki, S. Valentin, and D. Roggen, "The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices," *IEEE Access*, vol. 6, pp. 42 592–42 604, 2018.

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

[2] https://bitbucket.org/tudresden/mov-transport-dataset/

[3] https://www.bmvi.de/goto?id=344984