Grumiau, Christopher; Mostoufi, Mina; Pavlioglou, Solon; Verdonck, Tim

Article

Address identification using telematics: An algorithm to identify dwell locations

Risks

Provided in Cooperation with:
MDPI – Multidisciplinary Digital Publishing Institute, Basel

Suggested Citation: Grumiau, Christopher; Mostoufi, Mina; Pavlioglou, Solon; Verdonck, Tim (2020) : Address identification using telematics: An algorithm to identify dwell locations, Risks, ISSN 2227-9091, MDPI, Basel, Vol. 8, Iss. 3, pp. 1-12, https://doi.org/10.3390/risks8030092

risks

MDPI

Article

Address Identification Using Telematics: An Algorithm to Identify Dwell Locations

Christopher Grumiau [1], Mina Mostoufi [1,*], Solon Pavlioglou [1,*] and Tim Verdonck [2,3]

[1] Allianz Benelux, 1000 Brussels, Belgium; christopher.grumiau@allianz.be
[2] Department of Mathematics (Faculty of Science), University of Antwerp, 2000 Antwerpen, Belgium; tim.verdonck@kuleuven.be
[3] Department of Mathematics (Faculty of Science), Katholieke Universiteit Leuven, 3000 Leuven, Belgium
* Correspondence: mina.mostoufi@allianz.be (M.M.); solon.pavlioglou@allianz.nl (S.P.)

Received: 16 June 2020; Accepted: 21 August 2020; Published: 1 September 2020

Abstract: In this work, a method is proposed for exploiting the predictive power of a geo-tagged dataset as a means of identification of user-relevant points of interest (POI). The proposed methodology is subsequently applied in an insurance context for the automatic identification of a driver's residence address, solely based on his pattern of movements on the map. The analysis is performed on a real-life telematics dataset. We have anonymized the considered dataset for the purpose of this study to respect privacy regulations. The model performance is evaluated based on an independent batch of the dataset for which the address is known to be correct. The model is capable of predicting the residence postal code of the user with a high level of accuracy, with an f1 score of 0.83. A reliable result of the proposed method could generate benefits beyond the area of fraud, such as general data quality inspections, one-click quotations, and better-targeted marketing.

Keywords: telematics; address identification; POI; machine learning; mean shift clustering; DBSCAN clustering; fraud detection

1. Introduction & Relevant Work

Insurance activities are conducted on the basis of a truthful relationship between insured and insurer. Data completeness and reliability help to bolster this relationship. For instance, the residence region of the insured plays a major role in pricing. However, it is common that this information is not available, not correctly added to the database, or simply not current anymore. It has been observed that this lack of structure is sometimes enabling poor pricing and even occasionally hinting at the possibility of fraud. This paper presents an effort to mitigate this risk by exploiting the predictive power of a telematics dataset that can be leveraged in an insurance context.

Different machine learning methodologies are exploited in the literature in the context of telematics. Meiring et al. (2015), reviewed intelligent approaches for using telematics data in different applications. Moreover, Wuthrich (2017) utilized the k-means clustering method for categorizing the driving styles of drivers to enrich the insurance pricing scheme. In another application, Dong et al. (2016), by means of deep learning (RNN and CNN), identified the driving styles of drivers based on recorded GPS data. They presented a clustering-based ML approach for predicting the risk of the insured drivers based on their recorded GPS data as well as the acceleration and velocity in all directions.

On the topic of telematics being used as a means of predicting claims frequency and improving pricing, Pesantez-Narvaez (2019) examined the performance of XGBoost approach for predicting the claims based on a telematics dataset, including total annual distance driven and percentage of total distance driven in urban areas.

Telematics data in combination with the CNN algorithm has also been used for vehicle type identification by Nguyen (2018), based on data that were collected from the users' smartphones. Verbelen and Antonio (2018) combined the traditional insurance pricing with the telematics approach to include the driving behavior of the insured driver using generalized additive models and compositional predictors. Wang et al. (2017) implemented driver identification based on the telematics data and Random forest classification method. Qazvini (2019) compared the performance of a Poisson regression model and a zero-inflated Poisson (ZIP) model for the prediction of the claim frequency based on the telematics data that were collected from insured drivers.

The rich information coupled to big geo-tagged datasets has sparked many projects aiming to generate some level of structure out of unstructured information.

One such application is given here[1], where POI are automatically identified based on the spatial density of the pickup locations requested on a taxi platform. Their approach utilizes density-based models to automatically identify clusters of pickup locations where the density is higher. The main assumption is that clusters with a high density of points are to be considered more important.

Along the same lines, Yang et al. (2017) have also considered the density of the spatial distribution of points. Using the Laplacian zero crossing, they define the boundaries of what they consider an increased-interest region (POI), based on the locations of millions of geo-tagged Flickr photos. The added value of that approach is the simultaneous definition of the POI, as well as its physical boundaries.

Finally, the work of Deng et al. (2019) utilizes the same concept of spatial density to address the problem of the automatic identification of an urban center. In this work, the geo-tagged items to be clustered are POI, and their density is thought to represent the city center. By representing the POI density as a surface, similar to that utilized in terrain representation, the authors are introducing the concept of the surface contours of the density function. They subsequently utilize the contours to generate a contour tree, which hierarchically groups high-density regions together. The physical boundaries of the region of interest, are, here too, defined automatically.

A different source of geo-tagged data is telematics. By monitoring the motility patterns of an individual via specialized sensors, we are now able to generate vast amounts of geospatial data, rich in information and ready to be used for various life-improving projects. In one such application, McKenzie et al. (2019) utilize user-generated geo-content to help identify financial access points in sub-Saharan Africa.

The methods that are discussed above provide the background for this study; however, they present two shortcomings in the context of our application. Firstly, when the data richness is lower and the denser regions are scarce, fitting a density surface might prove problematic, whereas a straightforward clustering method (which is, however, density based) is the most efficient approach, without compromising on accuracy. In our application the median of the number of trips in the drivers' population is equal to 177; significantly lower than the vast volumes of geo-tagged pictures or available trips per user in the other studies. Furthermore, the aforementioned applications do not deal with competing classes of dense geo-tagged items. In our specific application, after identifying the denser regions (dwell locations), we also have to classify them as per the specific meaning that they carry within the context of the application, i.e. identify a destination as being a work address vs being a home address.

This paper proposes a methodology for the automatic identification and classification of a dwell location (or POI) of an individual. Subsequently, we apply the proposed methodology for the automatic identification of a user's residence address based on motility patterns that arise from a telematics dataset.

---

[1] https://blog.gojekengineering.com/fantastic-drivers-and-how-to-find-them-a88239ef3b29.

The remainder of this paper is organized, as follows: Section 2 provides a framework including (Section 2.1) the dataset utilized, (Section 2.3) how the datasets are preprocessed, and (Section 2.2) discussion on the used algorithms. In Section 3, (Section 3.1) the prediction results of clustering algorithms are presented and discussed, (Section 3.2) the performance of the exploited algorithms is evaluated, and (Section 3.3), (Section 3.4) the results of classification as residential address or not are presented.

2. Methodology

In this work, we take advantage of available telematics data in order to identify the user-relevant POI in the form of spatial clusters:

$$C_{ij} : \{d_k \;\; k = 1 \ldots n_{ij} | \;\; d_k \in C_{ij}\}, \tag{1}$$

$$\pi_{jl} : \{C_{oj} \;\; o = 1 \ldots m_j | \;\; C_{oj} \in POI_l\} \tag{2}$$

where C_{ij} is spatial cluster for the user j and destinations d_k's. Point of interest l for user j ,π_{jl}, is defined as set of spatial clusters C_o's relevant to intended point of interest POI_l.

Out of the defined user-relevant POI, a prediction is made of the particular POI that corresponds to the intended POI, POI_l, for instance, the residence address of the driver:

$$\pi_{prediction} : \{p_q \;\; q = 1 \ldots n | \;\; p_q \in POI_l\}, \tag{3}$$

where $\pi_{prediction}$ is set of POI considered for prediction, while p_q's are user's trips relevant to POI_l .

To this end, a single user is described by a point cloud of all the trips' end locations:

$$U_r, \;\; r = 1 \ldots n : \{p_s, \;\; s = 1 \ldots m\}, \tag{4}$$

where U_r is representing the r^{th} user out of n total users. On the other hand p_{rs} is the s^{th} trip of r^{th} user among all m total trips of user U_r.

Using the Mean Shift or DBSCAN clustering methods, the points are clustered into what we define as individual destinations. Here, the residential address is considered as intended POI. Indeed other types of POI can be also considered and this assumption does not reduce the level of generality of the methodology and it is just assumed for the purpose of methodology illustration.

We assume two types of datasets are available for carrying out the POI identification task. The first is a database of geo-tagged items, representing the motility pattern of the user in question. The second is a database of ground truth information about the exact location of the target POI. We have examined the ratio of dual-address drivers and we do not expect it to severely impact our results, as, in our portfolio, their proportion is quite low. This fact led us to the fundamental assumption that each user has only one true residential address.

In this work, we take advantage of available telematics data in order to identify the user-relevant POI in the form of spatial clusters. Out of the defined user-relevant POI, a prediction is made of the particular POI that corresponds to the residence address of the driver.

Figure 1 shows a schematic representation of the trip point cloud around the user's residence address. Both Mean Shift and DBSCAN clustering methods are tuned via a parametric study so that the result of the clustering is representative of a series of single destinations:

$$Clusters : \{d_t, \;\; t = 1 \ldots n\}, \tag{5}$$

where d_t's are single destinations attributed to $Cluster_w$'s ($w : 1 \ldots Number\ of\ clusters$) .

In the figure, after clustering, the points have been split in two clusters with their respective centroids represented by an orange circle. Each centroid acts as a single-point representative of the whole cluster and it is checked for its proximity to the real home address:

$$Clusters : \{Center_y, \ y = 1 \dots n\}, \tag{6}$$

where $Center_y$'s are identified centers for $Cluster_w$'s ($w : 1 \dots Number\ of\ clusters$).

For the training and evaluation of the model, a circle is defined per user (user r), with radius r_{home}, and its center positioned on the exact coordinates of the user's actual address (X_r). The prediction is considered to be successful when the centroid of the cluster (K_y) that is predicted as a home address lies within this circle ($|K_y - X_r| < r_{home}$).

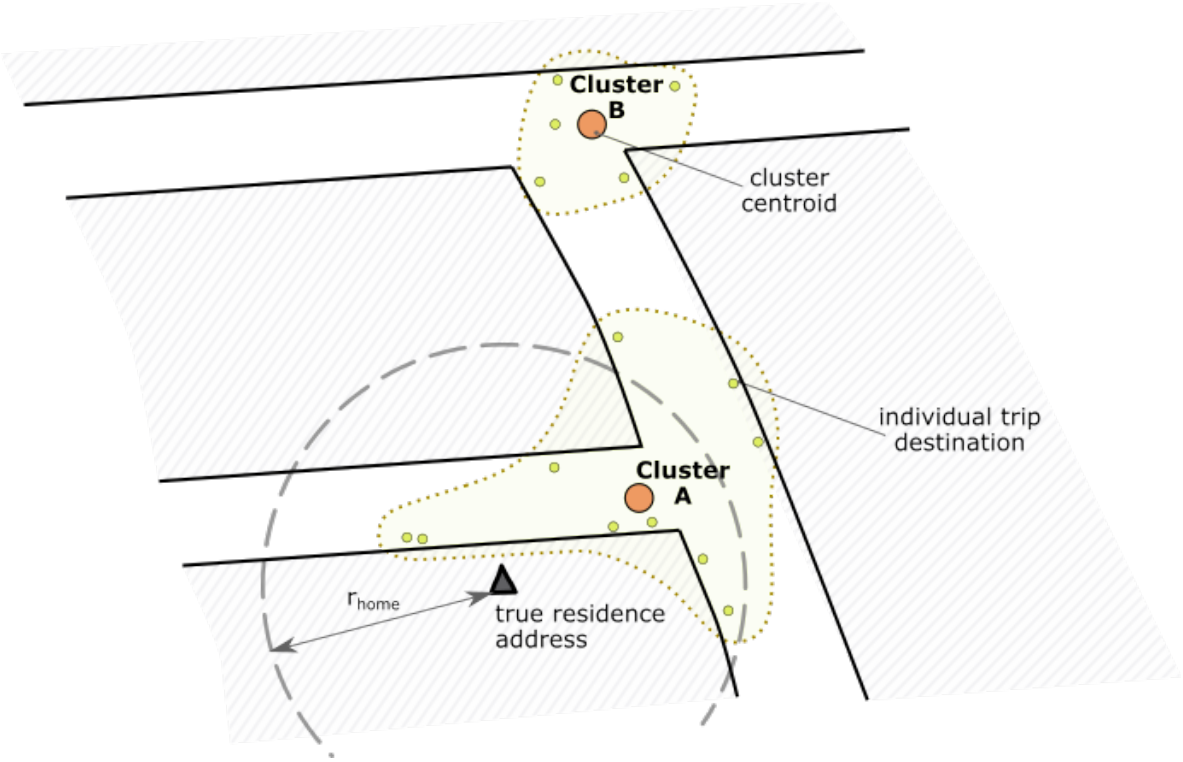

Figure 1. Methodology Explanation.

In the case of Figure 1, cluster A is considered to be a positive observation, while cluster B is a negative one. In Section 3, we address the effect of this radius on the prediction accuracy.

Each destination cluster is subsequently enhanced with information that describes its properties. The information introduced about the cluster is descriptive of the user's interaction pattern with it. The extracted/created specific features are introduced and discussed in Section 2.3.

Following the feature engineering phase, a classification model is constructed with the purpose of identifying the destination that is the most likely to be the residence address of the individual under examination.

2.1. Dataset

The dataset utilized for the analysis has been extracted from an in-house telematics database. The information in the database has been recorded utilizing SMAAS (smartphone as a sensor). The dataset has a total of 1438 individuals and 525,663 trips.

Each user was monitored throughout a certain time interval, while his location was recorded at a frequency of around 1/3 Hz. For the purposes of this analysis, the dataset is processed and reorganized so that the intermediate points are dropped and only the start and end locations of the

user's trips remained in the dataset. In addition to the location data, temporal information on arrival and departure are also included in the dataset. Another dataset containing the address of the user is utilized in combination with the start/finish trip locations.

During a data cleaning step, we made sure that we removed from the training dataset all users who, throughout their trips, have not once arrived at a distance smaller than or equal to 500 m from their true registered address. As such, the useful dataset was reduced to 928 individuals with a corresponding total of 371,258 trips, for further processing.

2.2. Clustering Algorithms

Two different clustering methods, DBSCAN and Mean Shift , are employed for the identification of the individual destinations of the user. In the following paragraphs, we elaborate on these two algorithms and their theoretical backgrounds, and present a performance comparison.

The DBSCAN (Density Based Spatial Clustering Applications with Noise) algorithm is a robust clustering approach for a dataset mixed with noise, as indicated in Chakraborty al. (2011) and Tran et al. (2013). This method utilizes two independent parameters for detection of the body and border of a cluster of an arbitrary shape, solely based on the density of the points within it. The algorithm works well in detecting regions of higher density, Zhong et al. (2019).

However, it presents a few problems in the context of user-relevant POI detection. This algorithm only works well for clusters of similar density, Wang et al. (2019). Furthermore, DBSCAN tends to merge different clusters into a single one, regardless of the shape and size of the resulting cluster, if a chain of equal density points exists as a bridge between them. This property, albeit necessary and desired in other applications, presents a problem in the context of POI detection, as it allows for regions of interest that are arbitrarily large, which inherently contradicts the notion of a POI.

The most important drawback of this algorithm is its sensitivity to the model parameters, as indicated in Ren et al. (2014). Parameter optimization is necessary to assure that the clustering performance is not negatively affected by the selected model parameters. Despite the possible complications of the algorithm, it has been tried with success, also because of its ability to exclude points that are considered to be noise. The DBSCAN algorithm has the ability to handle the noisy data efficiently, even in a dynamic environment with an ever-changing dataset, Chakraborty al. (2011).

The non-parametric Mean Shift algorithm is based on the assumption that each dense region represents a cluster. This approach to clustering allows the method to be independent of further assumptions regarding the features of the clusters, such as the number of clusters and their distribution.

The method shifts the kernel centroid iteratively towards the direction of the average of the points contained in the kernel. The difference between subsequent centroids defines the mean shift vector, which always points towards the direction of maximum increase in the density. The governing parameter of the Mean Shift algorithm, the bandwidth parameter, is the radius of proximity that is considered by the kernel function, which is adjustable based on the application.

The governing parameters of the algorithm are interpretable, and the performance is stable. Despite its potential for improved clustering accuracy, the Mean Shift approach is a computationally expensive method for very large datasets. In this work, the dataset size does not impose a high computational burden, thus computational efficiency is not a governing performance criterion.

The Mean Shift algorithm is very sensitive to the selection of the bandwidth parameter (h). Thus, the bandwidth parameter needs to be chosen carefully based on the application. Inappropriate adjustment of h can lead to an incorrect number of clusters or an undesirable cluster configuration.

2.3. Feature Engineering

The dataset undergoes three preparation steps before any meaningful prediction can take place. First, the user's trips are filtered, such that only the last known location of the trip remains as a representative of the trip's destination. Subsequently, the total population of ending locations is segmented into individual destinations, such that the dataset reduces in size to the total number

of different unique destinations. A single point represents the population of each destination, namely, the coordinates of the centroid of the identified cluster. Finally, for every separate cluster, a feature-engineering step is necessary in order to enrich the knowledge about the user interaction with the cluster. Table 1 presents the list of covariates that are generated .

Table 1. List of generated features that constitute properties of an individual destination/cluster of the user's totality of trips.

Feature	Description
Cluster size	Absolute number of user visits to this destination
Average time of arrival week	Time of the day (in min) that the user visits the destination on average; for visits that happen during the week
Average time of arrival weekend	Time of the day (in min) that the user visits the destination on average; for visits that happen during the weekend
STD time of arrival week	Standard deviation (in min) of the time of arrival at this destination; for visits that happen during the week
STD time of arrival weekend	Standard deviation (in min) of the time of arrival at this destination; for visits that happen during the weekend
Average stay	The amount of time (in min) that the user spends at the destination before the next departure
STD stay	Standard deviation of the amount of time (in min) that the user spends at the destination before the next departure
Freq trips week	Absolute number of user visits to this destination during the week
Freq trips weekend	Absolute number of user visits to this destination during the weekend
Covariance lat/lon	Covariance of the latitude and longitude of all exact stopping points within the cluster. It is a measure of how scattered the visits are within what was deemed a single destination

To summarize the above, a concise representation of the methodological steps is presented in Figure 2.

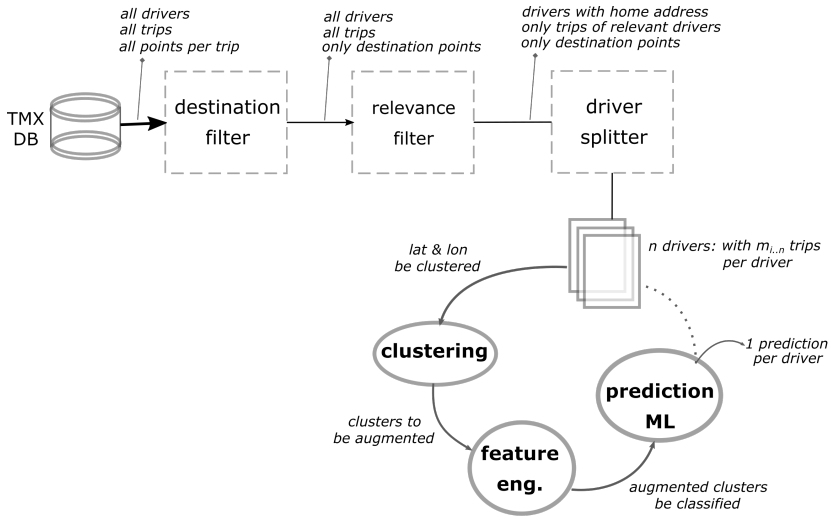

Figure 2. UML of methodological steps.

3. Performance Comparison & Results

3.1. Clustering Performance Comparison

Following the feature-engineering phase, a prediction is made using one of the unsupervised machine learning approaches based on the observed feature space presented in Section 2 (Section 2.3).

In this section, the achieved prediction accuracy of the two different clustering techniques is compared. When considering the imbalanced nature of the dataset, the Precision-Recall curve of the positive class is considered as the main performance criterion. A high area under the Precision-Recall curve is desired, representing a high level of both precision and recall. A metric of this property is the f1 score of the model, which will also be presented in the results.

For the considered clustering techniques, a parametric study is carried out to investigate the effect of the independent model parameter on performance, while the home distance tolerance, r_{home}, is kept constant and equal to 500 m. The value of 500 m is selected, such that all intended applications of the method are applicable without ambiguity. The effect of the r_{home} selection is also addressed in an effort to assess the robustness of the method for business applications that require potentially higher spatial precision.

3.1.1. Tuning DBSCAN

For DBSCAN, the influence of the model parameter, *Eps*, has been examined within a range from 0.0005 to 0.05, with the obtained f1 score presented in Figure 3. These *Eps* values roughly correspond to a physical radius that is in the range of 35 m to 350 m. The second hyper-parameter of DBSCAN has been chosen to be equal to three.

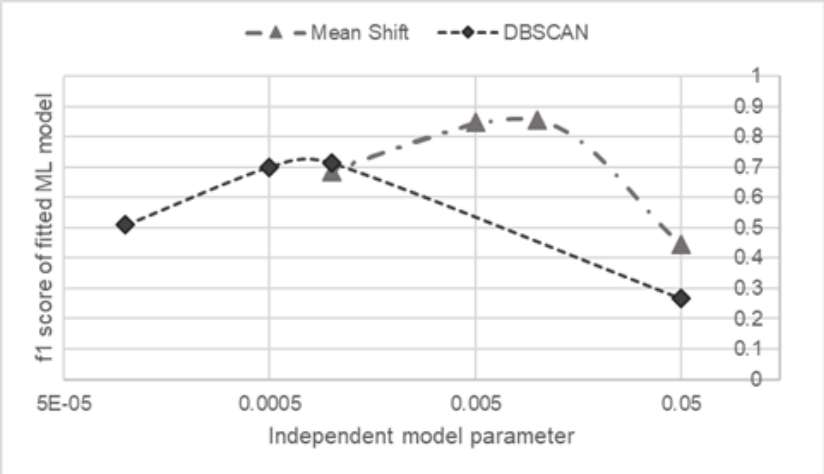

Figure 3. Fitting the hyper-parameter values for Mean Shift and DBSCAN.

For given *Eps* and physical radius, the algorithm will keep expanding the cluster as long as the three or more close neighbors are found within distance *Eps*. A too small radius *Eps* will lead to many destinations being excluded as being noise, while a too large value for the same radius runs the danger of joining destinations that should have ideally been separated and giving the cluster an irregular shape. The latter will transfer the centroid of the combined cluster far from the true home address. Moreover, the types of points that will end up being clustered will be very diverse, which will generate features that are of limited predictive power.

3.1.2. Tuning Mean Shift

The same study is performed for Mean Shift, when considering values for the *h* parameter in the same range. A change in this radius would mean that the size of the control circle is affected. Just as

with DBSCAN, a very large radius would lead to clustering of distant locations as a single destination, and the centroid would drift from the ideal, close-to-home location.

One difference of Mean Shift is that the area of influence of a single cluster is predetermined, and equal to the circle in question. The method is not allowed to append points that are in close proximity to the generated cluster, as DBSCAN would. Instead, it will generate a new destination with its own points, which is h distance away from the former cluster.

Given the nature of the problem that we are trying to solve, it appears that Mean Shift can more robustly cluster the individual trips to different destinations. In the extreme case where the parameter value is too low, the total number of clusters will be equal to the individual trips made by the user, in which case the feature engineering process that is mentioned in Section 2 (Section 2.3) would not provide any added value.

Following the hyper-parameter tuning of the two clustering methods, the best-performing setup from each model is presented for comparison in Figure 4. The superior performance of Mean Shift is notable, with a PR curve obtaining recall rates in the order of 85% and a precision of 80%. Please define if appropriate.

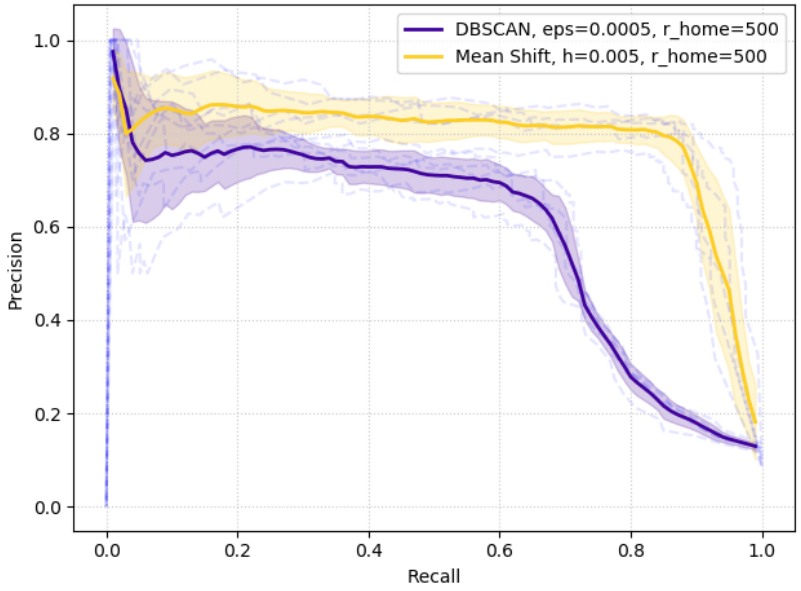

Figure 4. PR curves for best-performing setup from Mean Shift and DBSCAN.

3.2. Spatial Accuracy Investigation

Finally, the possibility to improve on the spatial accuracy of the prediction is assessed. A parametric study is performed, varying the radius r_{home}. For smaller values of the radius, a more exact prediction of the home address is obtained. It is logical that the model would only be accurate within a certain distance tolerance given the use of movement data as a proxy for the definition of true address. It is therefore expected that any improvement on the spatial accuracy will come at the expense of some of the predictive capacity of the model.

The best performing algorithm, Mean Shift, is selected for the analysis. The model specific parameter is kept constant and equal to 0.005 (≈200 m) and the model fitting is repeated for home distance tolerance parameters (r_{home}) of 500 m, 200 m, and 100 m. The results are depicted in Figure 5.

As expected, a drop in the predictive capacity is observed for smaller tolerances (meaning higher spatial accuracy). However, it is notable that, even in the most demanding scenario with a prediction accuracy of just 100 m, the model performs relatively well, providing an educated guess with an

f1 score of 0.744. In all cases, the model performs much better than the "no-skill" prediction that represents randomness (see Figure 5).

The simulations are performed on PC equipped by Intel Core i7-8750H 2.20 GHz CPU and 32 GB RAM.

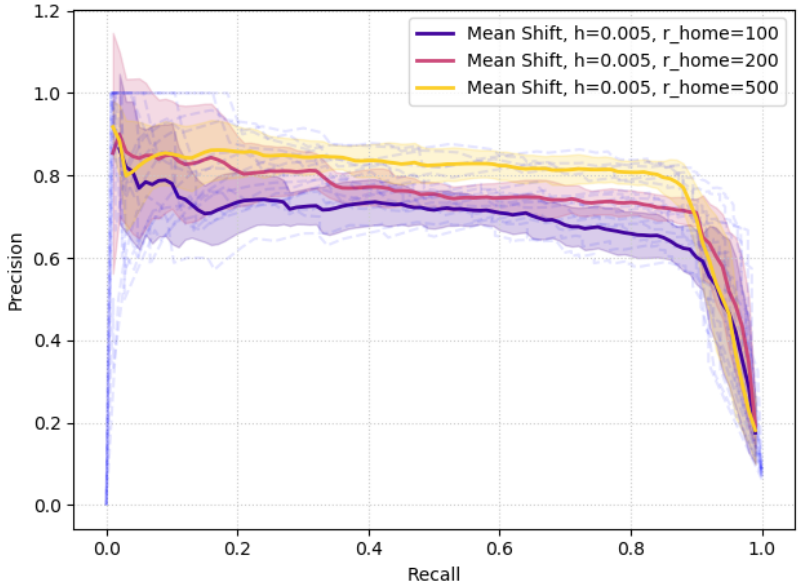

Figure 5. Comparison of the performance of the prediction for different values of the home distance tolerance, r_{home}.

3.3. Rule-Based Prediction

A simple rule is devised for the 'discovery' of the home address of every individual, out of the total set of destinations from the recorded trips. For each person, the destination that has been visited most frequently will be selected as the residence address. The approach implies that every individual of the dataset is expected to have exactly one residence address. This is in accordance with the data cleaning process indicated in Section 2 (Section 2.1). The method performs relatively well with an achieved f1 score of 0.785. This result was used as a baseline and in the next section it is compared to machine learning-based methodologies.

3.4. Machine Learning Prediction

For the more advanced modeling, the problem is represented as a binary classification problem, where the model aims to classify every individual destination cluster of the user as either being a home address or not (unity or zero labels respectively). The cluster classification phase (as being a residence address or not) is performed using one linear and two non-linear methods, namely, Logistic Regression, Random Forest and Multilayer Perceptron. The results are subsequently summarized and compared with the rule-based approach presented in Table 2. The analysis indicates that the three ML models perform at a comparable level to each other, but outperform the rule-based prediction by about 4%. The increased performance of the ML models is justified by the utilization of all 10 generated features made available in the feature engineering phase. In contrast, the rule-based prediction only looks at the relative cluster size before classifying the destination.

Table 2. Comparison of different prediction methodologies (Rule-based approach, Logistic Regression, Random Forest, Multilayer Perceptron) on the same set (test set).

	Precision	Recall	f1-Score	Support
Rule	0.735	0.844	0.786	162
LR	0.774	0.878	0.824	162
RF	0.794	0.857	0.825	162
MLP	0.795	0.859	0.826	162

The prediction appears to be very sensitive to the clustering methodology used and a lot less to the classification method. This indicates that the first phase of modeling is the most important aspect of the proposed methodology for user-relevant POI identification, namely the clustering of the individual trips into meaningful destinations. The importance of the correct selection of the clustering method becomes paramount given the sensitivity of the method in the clustering phase. Mean Shift has proven that, despite DBSCAN being preferred for spatial clustering in literature, it can perform better and bring an uplift in the f1 value in the order of 17%.

Given the similarity of the performance and for the sake of simplicity, only the Multilayer Perceptron(MLP) results will be demonstrated and discussed in this section. The considered MLP was tuned to comprise two hidden layers of 30 and 15 nodes respectively. The model is solved using the 'Adam' optimizer. The dataset is split into train and test sets, chosen at 80% and 20% of the total observations, respectively, for which the full set of available features is included. Using five-fold cross validation, the fitting was performed multiple times, such that potential bias in the presented results would be addressed. The uncertainty bandwidth accompanying the presented curves accounts for the standard deviation in the results of each of the five model runs.

The model itself is comprised of two sequential steps. In the first step, the prediction of the machine learning model returns a probability vector for the total number of clusters for all the users in the dataset. This is succeeded by a logical check where the probabilities are converted to binary values, taking into account the fact that only a single home address will correspond to each user. Here, too, the assumption is in line with the data cleaning process that is indicated in Section 2 (Section 2.1).

The obtained performance level for the configured MLP algorithm is given in Tables 3 and 4.

Table 3. Performance of the MLP Model for a representative out of the five folds (test set).

	Precision	Recall	f1-Score	Support
negative	0.98	0.98	0.98	1442
positive	**0.81**	**0.85**	**0.83**	166
accuracy	-	-	0.96	1608

Table 4. Confusion matrix of the MLP model for a representative out of the five folds (test set).

	Negative (Data)	Positive (Data)
Negative (Prediction)	1408	34
Positive (Prediction)	25	141

4. Conclusions

In this paper, we utilize the DBSCAN and Mean Shift clustering methods in order to identify the residential address of subscribed drivers. A parameter study is performed for fine-tuning the model parameters and home distance tolerance. Both the Mean Shift and DBSCAN approaches perform well, with f1 scores of 0.66 and 0.83, respectively. Furthermore, it is shown that clustering trips with Mean Shift facilitates predictions that outperform the ones based on DBSCAN clustering. Finally,

a supervised ML approach using Logistic Regression, Random Forest, and Multilayer Perceptron is implemented for the identification of the user home residential address. The model performance measures show that the ML-based address identification approach yields reliable predictions with a precision of 81% and a recall of 85%.

The methodology in this paper is demonstrated for the identification of the home address. However, we believe it to be suitable for the identification of any kind of user-relevant POI, as long as ground truth data exist for the algorithm to train on.

The proposed model could work very well within an insurance context as a fraud prevention or identification mechanism, by contrasting the predicted addresses to the registered ones. It would also be a source of useful information at a primitive stage in the acquisition of new contracts, as it would facilitate marketing, and automatically populate part of the questionnaire that is necessary at the subscription phase. The latter is also known in an insurance context as a one-click policy.

The concept is promising and additional research on this topic could result in predictions that are even more accurate. Other clustering methodologies (like OPTICS) in combination with a more elaborate feature-engineering phase could possibly render even better predictions. In this work, the utilized ML model is deemed to be sufficient to demonstrate the predictive capacity of the concept. A thorough examination of different ML techniques would potentially result in an uplift to the achieved accuracy.

Author Contributions: Methodology and simulations are conducted by M.M. and S.P. and the original draft is prepared by all authors. Reviewing and editing are performed by all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Allianz Benelux and the Allianz Chair on Prescriptive Business Analytics in Insurance at KULeuven.

Acknowledgments: We are grateful for the contribution of the Allianz Chair on Prescriptive Business Analytics in Insurance at KULeuven for enriching the contents of the paper. Furthermore, we appreciate the opportunity that Allianz Benelux provided for us to dedicate time to R&D activities during our work hours.

Conflicts of Interest: The authors declare no conflict of interest.

References

Deng, Yue, Jiping Liu, Yang Liu, and An Luo. 2019. Detecting urban polycentric structure from POI data. *ISPRS International Journal of Geo-Information* 8: 283. [CrossRef]

Dong, Weishan, Jian Li, Renjie Yao, Changsheng Li, Ting Yuan, and Lanjun Wang. 2016. Characterizing Driving Styles with Deep Learning. *arXiv*, arXiv:1607.03611.

McKenzie, Grant, and R. Todd Slind. 2019. A user-generated data based approach to enhancing location prediction of financial services in sub-Saharan Africa. *Applied Geography* 105: 25–36. [CrossRef] [PubMed]

Meiring, Gys Albertus Marthinus, and Hermanus Carel Myburgh. 2015. A Review of Intelligent Driving Style Analysis Systems and Related Artificial Intelligence Algorithms. *Sensors* 15: 30653–82. [CrossRef] [PubMed]

Nguyen, Linh. 2018. A Vehicle Classification Algorithm Based on Telematics Data. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Pesantez-Narvaez, Jessica, Montserrat Guillen, and Manuela Alcañiz. 2019. Predicting Motor Insurance Claims Using Telematics Data—XGBoost versus Logistic Regression. *Risks* 7: 70. [CrossRef]

Qazvini, Marjan. 2019. On the Validation of Claims with Excess Zeros in Liability Insurance: A Comparative Study. *Risks* 7: 71. [CrossRef]

Ren, Yazhou, Uday Kamath, Carlotta Domeniconi, and Guoji Zhang. 2014. Boosted Mean Shift Clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Berlin and Heidelberg: Springer, vol. 8725, pp. 646–61. [CrossRef]

Chakraborty, Sanjay, Naresh Nagwani, and Lopamudra Dey. 2011. Performance Comparison of In-cremental K-means and Incremental DBSCAN Algorithms. *International Journal of Computer Applications* 27: 14–18. [CrossRef]

Tran, Thanh, Klaudia Drab, and Michal Daszykowski. 2013. Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems* 120: 92–96. [CrossRef]

Verbelen, Roel, and Katrien Antonio. 2018. Unraveling the Predictive Power of Telematics Data in Car Insurance Pricing. *SSRN Electronic Journal*. [CrossRef]

Wang, Bo, Smruti Panigrahi, Mayur Narsude, and Amit Mohanty. 2017. *Driver Identification Using Vehicle Telematics Data*. *SAE Technical Paper*. [CrossRef]

Wang, Xiaochun, Xiali Wang, and Don Mitchell Wilkes. 2019. *Machine Learning-based Natural Scene Recognition for Mobile Robot Localization in An Unknown Environment*. Berlin and Heidelberg: Springer.

Wuthrich, Mario. 2017. Covariate selection from telematics car driving data. *European Actuarial Journal* 7. [CrossRef]

Yang, Yiyang, Zhiguo Gong, Qing Li, Leong Hou U, Ruichu Cai, and Zhifeng Hao. 2017. A robust noise resistant algorithm for POI identification from Flickr data. Paper presented at the IJCAI International Joint Conference on Artificial Intelligence, Melbourne, Australia, August 19–25, pp. 3294–300. [CrossRef]

Zhong, Li, Tiantian Zhang, and Bo Yuan. 2019. A critical note on the evaluation of clustering algorithms. *arXiv*, arXiv:1908.03782.