Crash Prediction and Risk Assessment with Individual Mobility Networks

Riccardo Guidotti
University of Pisa, Italy
riccardo.guidotti@di.unipi.it

Mirco Nanni
ISTI-CNR, Pisa, Italy
mirco.nanni@isti.cnr.it

***Abstract*—The massive and increasing availability of mobility data enables the study and the prediction of human mobility behavior and activities at various levels. In this paper, we address the problem of building a data-driven model for predicting car drivers' risk of experiencing a crash in the long-term future, for instance, in the next four weeks. Since the raw mobility data, although potentially large, typically lacks any explicit semantics or clear structure to help understanding and predicting such rare and difficult-to-grasp events, our work proposes to build concise representations of individual mobility, that highlight mobility habits, driving behaviors and other factors deemed relevant for assessing the propensity to be involved in car accidents. The suggested approach is mainly based on a network representation of users' mobility, called Individual Mobility Networks, jointly with the analysis of descriptive features of the user's driving behavior related to driving style (e.g., accelerations) and characteristics of the mobility in the neighborhood visited by the user. The paper presents large experimentation over a real dataset, showing comparative performances against baselines and competitors, and a study of some typical risk factors in the areas under analysis through the adoption of state-of-art model explanation techniques. Preliminary results show the effectiveness and usability of the proposed predictive approach.**

***Index Terms*—Mobility Data Model, Crash Prediction, Individual Mobility Network, Mobility Data Mining, Car Insurance**

I. Introduction

The huge availability of mobility data collected by car telematics and car insurance companies is typically used to provide to end-users services like pay-as-you-drive contracts, anti-theft control, and prompt emergency rescue in case of accidents [1]. However, a fundamental task of car insurance companies is to find the most appropriate policy pricing for a customer, which consists of a trade-off between profit and competitiveness. In this context, risk assessment is probably the most critical problem addressed.

The most intuitive way to solve the risk assessment problem is to estimate the customer's risk of having accidents in the near future [2] since high-risk ones are likely to cause the company a loss (paying the costs of her accidents), while low-risk ones are more likely to provide a plain profit. The basic objective is not only to recognize the real risk level of a customer but also to understand possible causes [3]. Hence, we aim to reach two distinct results. First, predicting the customer's risk score: given a car insurance customer, provide a risk score relative to the near future, e.g., the next year or the next month. We expect this estimate to be much dependent on how the customer drives, as well as on the conditions of the surrounding environment [4]–[6]. Accordingly, the methodology we propose is based on the computation of individual driving features, describing how much the user drives and how much dynamically, also related to the general characteristics of mobility in the places that the user visits. The second result we pursue is to infer risk mitigation strategies: given a car insurance customer and her risk score, we would like to identify the characteristics of her driving behavior [7] that determine her risk score. From a prescriptive viewpoint, this is going to provide to the customer indications of how to lower down her risk score, with benefits for her (in terms of safety and insurance costs) and the insurance company (in terms of costs for accidents). The approach under investigation queries the predictive models adopted for understanding which features decided for the prediction [8].

Since raw mobility data collected by car telematics and car insurance companies is limited to positions and events of the vehicle [1] with no vision of what happens around it, or further structured and complex information, in order to achieve our goals we need to augment the individual data with additional knowledge. Indeed, raw mobility data describes elementary events (position, acceleration, etc.), while any modeling requires a higher-level vision of what is happening to the user. That should provide some clear semantics, e.g., some typical maneuvers that involve sequences of deviations, sudden decelerations, etc. Recognizing and making them explicit is expected to be an important need. A specific type of semantics is related to the meaning that the different parts of the mobility have for the individual: recurrent vs. systematic trips [9], frequent locations vs. single visit ones [10], transit locations vs. long stays [11], etc. To infer this type of information, we need to model the mobility of the individual as a whole, creating a single, complete picture of it. To this aim, we adopt Individual Mobility Networks (IMNs) [12], [13], a network-based representation that integrates important locations, movements, and their temporal dimension in a succinct way. Such a model allows several different types of inference (detecting the purpose of the trip [12], simulating realistic mobility agendas [13], etc.), in contrast to others that are tailored around specific objectives. In particular, we exploit the integration of information in the IMNs formalism for inferring mobility indicators useful for the predictive/prescriptive purposes of the crash prediction task.

Crash risk means probability of accidents, which are statistically rare events [14]. This, together with the lack of a clear set of predictive indicators to adopt, make the risk prediction a very difficult task. Therefore, the proposed approach takes into account several different aspects: *individual* components of the driving behavior including those that can be derived from IMNs, elements considering the *collective* mobility of other users, and static *contextual* information such as road categories and the presence of points of interest.

Achieving a good prediction accuracy often conflicts with the understandability of the predictive model. In difficult settings, complex predictors such as deep neural networks can achieve better performances than simpler ones like decision trees, Bayesian classifiers, etc., yet, the formers are usually not human understandable [15]. We aim not only to provide good predictors for the car crash application but also extracting risk mitigation guidelines for the user, which means we are interested in understanding which factors made a driver a risky one in order to propose changes in her behavior that can reduce the risk. While that makes simpler models more appealing, we also explore methods for "explainable AI" [3], [16], aimed to extract explanations from not interpretable predictors. Finally, since the various individual mobility models and predictors are expected to be highly dependent on the geographical area under study, we aim to test the *transferability* of the models obtained through our approach from a region to another one, which provides first insights for tackling a sort of geographical instance of the general transfer learning problem [17].

We evaluate the proposed methodology on a dataset of real cars moving in three different areas, namely two cities (Rome and London), and one region (Tuscany, Italy). The results show that the individual mobility-based and context-aware approach we proposed improves performances over basic solutions that use state-of-art features; also, the analysis of predictions with explainable AI methods (in particular [18]) reveals that, indeed, most of the main factors that lead the models to decide for the riskiness of users belong to the newly introduced features.

The rest of the paper is organized as follows. Section II summarizes related work on crash prediction and individual mobility data models. In Section III we recall IMNs and further concepts for understanding the interpretable model designed for crash prediction described in Section IV. Section V presents experiments in the form of a case study in which we employ the proposed methodology. Finally, Section VI concludes the paper and discusses next challenges.

II. Related Work

The existing literature addresses the problem of crash prediction from various perspectives, but, at the time of writing, to the best of our knowledge, there are no existing works exploiting mobility data analysis and user modeling for crash prediction and risk assessment. Indeed, a large body of works focuses on real-time prediction of individual crashes, i.e., try to identify the events that lead to a crash in the next few seconds, thus providing feedback to the user as she drives [19]. In [20] is developed a model for real-time collision detection at road intersections by mining collision patterns. Similarly, but using different data, [5] tries to relate crashes to both behavioral characteristics and physiologic parameters. Other approaches work on identifying areas that show characteristics usually associated with accidents, such as increased traffic density, adverse weather conditions, etc., e.g., [4], [21], [22]. Besides features describing areas, the work in [23] also used individual vehicular data of cars (speed and time headway) passing through predefined detector stations for improving the performance of a probabilistic model. In [24] it is presented a review of the key issues associated with crash-frequency data as well as strengths and weaknesses of similar methodological approaches. While extremely useful, such approaches result in being not applicable to fields like car insurance, where the focus is in creating a general risk profile of the user, thus implicitly involving the prediction of her crash risk in the long run, such as few months in the future. Only a few, preliminary works are available in this direction, e.g., [2] adopts machine learning methods to predict driving risk on the basis of simplistic features describing the users' driving behaviors.

In order to improve the state-of-the-art in crash prediction, the proposal of this paper takes into account several various aspects, ranging from the driving behavior of the user to the types of environment she usually traverses. In the following, we briefly review the existing individual mobility data model. In [7], the notion of *mobility profile* is introduced, which summarizes the regular movements of a user. Such individual models are exploited in [25] for building an effective individual and collective movement predictor. The work in [12] provides a first definition of *Individual Mobility Networks* (IMNs), a network-based representation that integrates locations, movements, and their temporal dimension in a succinct way. In contrast to other models that are tailored around very specific objectives, IMNs have large applicability and allow several different types of inference: detecting the purpose of trips [12], simulating realistic mobility agendas [13], etc. In this work, we aim at exploiting such kind of models, in particular, IMNs for simultaneously integrating as much information as possible in a single formalism and inferring from it mobility indicators useful for predictive and prescriptive purposes.

An important collateral point is that the individual mobility models and crash predictors are expected to be highly dependent on the specific geographical area under study. For instance, it has been empirically verified that the trip purpose classifiers in [12] work very well in the geographical area where they were extracted, but their performances dramatically degrade if applied to areas with different characteristics. At the same time, not all areas are equally well covered by data, due to the non-homogeneous penetration of tracking devices, making it challenging to build different models for different areas from scratch. All this calls for methodologies that make it possible to adapt models built in data-rich areas to less rich ones, basically a geographical instance of the general transfer learning problem [17]. The experiment section of this work also includes preliminary results in this direction.

III. Setting The Stage

In the following, we introduce the definitions of *trajectory* [7] and *individual mobility network* [12], [13], useful for understanding the rest of the paper. We adapt them to the problem we are facing and the approach designed to solve it.

Definition 1 (Trajectory): A trajectory *is a sequence* $t = \langle p_1, \ldots, p_n \rangle$ *of spatio-temporal points, each being a tuple* $p_i = (x_i, y_i, z_i)$ *that contains latitude* x_i*, longitude* y_i *and timestamp* z_i *of the point. The points of a trajectory are chronologically ordered, i.e.,* $\forall 1 \leq i < n : z_i < z_{i+1}$.

Given a trajectory t we refer to its i-th point p_i with the notation $t[i]$, and to its number of points with $t.n$. Also, we indicate the longitude, latitude and timestamp components of point $t[i]$ respectively with the notation $t[i].x$, $t[i].y$, and $t[i].z$.

Definition 2 (Individual History): Given a user u*, we define the* individual history *of* u *as the set of trajectories* $H_u = \langle t_1, \ldots, t_n \rangle$ *traveled by* u*. Also, we denote with* $H_u^{[a,b]}$ *the subset of trajectories of* H_u *that occur in time interval* $[a, b]$*, i.e.* $H_u^{[a,b]} = \{t \in H_u \mid [t[1].z, t[t.n].z] \subseteq [a, b]\}$.

Given the individual history H_u of user u, we can extract from H_u the *individual mobility network* (IMN) G_u. An IMN describes the individual mobility of a user through a graph representation of her locations and movements, grasping the relevant properties and removing unnecessary details.

Definition 3 (Individual Mobility Network): Given a user u*, we indicate with* $G_u = (L_u, M_u)$ *her individual mobility network, where* L_u *is the set of nodes and* M_u *is the set of edges. Given an aggregation operator* agg*, for each node* $l \in L_u$ *we define the following functions:*

- $\omega(l) =$ *number of trips in* H_u *reaching location* l*;*
- $\delta(l) = agg(\{$*durations of stops in* $l\}$*);*
- $\rho(l) = agg(\{$*arrival times of trips reaching* $l\}$*);*
- $\pi_t(l) = agg(\{$*durations of trips reaching* $l\}$*);*
- $\pi_d(l) = agg(\{$*lengths of trips reaching* $l\}$*);*

Operator agg *can return either a single value (e.g. median) or a n-ple (e.g. average and standard deviation, or quartiles). The same functions are also defined on edges (movements)* $m = (l_i, l_j) \in M_u$ *in a similar way, this time considering only trips that start from* l_i *and reach* l_j*.*

Nodes in L_u are locations that represent a group of stop points, and similarly edges in M_u are movements that represent groups of similar trips between two locations. The computation of an IMN G_u starts from the history H_u of user u, obtaining the locations L_u through a spatial clustering-based aggregation of stop points (in particular, the TOSCA algorithm [10]) and a trajectory clustering of the trips between any pair of locations [13].

IV. Car Crash Prediction

In this section we formalize the car crash prediction problem, describe the methodology developed to solve it by means of individual, collective, and contextual features of the user's driving behavior.

A. Problem Formulation

We define the crash prediction problem in terms of associating with the recent historical mobility of a user the probability of having an accident in the next time period. The duration of the user's history to consider and of the next time period for which we make predictions are two fixed parameters. As mentioned in the introduction, reasonable durations for the context at hand will have the scale of one or more months.

Definition 4 (Crash Prediction and Risk Assessment): Given the following three parameters: prediction time τ_p*,* history depth τ_h *and* prediction span τ_s*, we define the two time intervals* $\bar{z_p} = [\tau_p - \tau_h, \tau_p]$*, named* predictors interval*, and* $\bar{z_t} = (\tau_p, \tau_p + \tau_s]$*, named* target interval*. Then, the* crash prediction problem *consists in evaluating if user* u *will have a car crash during period* $\bar{z_t}$ *and what is the crash probability, based on the analysis of the user's mobility during period* $\bar{z_p}$*. More formally, we want to estimate:*

$$p_{crash}(u) = P(u \; has \; crash \; in \; \bar{z_t} \mid H_u^{\bar{z_p}})$$

The period $\bar{z_p}$ represents the knowledge we have about the user at the moment of assessing her risk, while $\bar{z_t}$ is where the crash to predict will or will not happen.

B. General approach

Our approach consists in approximating the probability $p_{crash}(u)$ in our problem definition in two steps: *(i)* first, the knowledge contained in $H_u^{\bar{z_p}}$ is represented through a set of meaningful yet (necessarily) lossy features, that will be discussed in details in the next sections; then, *(ii)* the probability function is learned through data-driven models, in particular, standard machine learning predictors.

User's features. We model each user u with a vector of m features computed over her predictors interval: $x_u^{\bar{z_p}} = \langle f_1, f_2, \ldots, f_m \rangle$. We name $X^{\bar{z_p}} = \langle x_1^{\bar{z_p}}, x_2^{\bar{z_p}}, \ldots, x_n^{\bar{z_p}} \rangle$ the matrix of n vectors describing the behavior of n users. We indicate with $y^{\bar{z_t}}$ the vector saying if a user has experienced a crash in the target interval $\bar{z_t}$, i.e., $y_u^{\bar{z_t}} = 1$ if user u had a car crash in period $\bar{z_t}$, $y_u^{\bar{z_t}} = 0$ otherwise.

Machine Learning models. Given $X^{\bar{z_p}}$ and $y^{\bar{z_t}}$, we train a machine learning classifier and we obtain as output a car crash predictor function $p_{crash}(\cdot)$. The crash predictor takes as input a vector $x_u^{\bar{z'_p}}$, describing user u's mobility in a given predictors interval $\bar{z'_p}$, and returns the probability she will have a crash in the corresponding target period $\bar{z'_t}$, based on the training performed on $X^{\bar{z_p}}$ and $y^{\bar{z_t}}$.

As machine learning classifiers we considered several possible options, including k-NN classifiers, Naive Bayes, decision trees, support vector machines, feed-forward neural networks, random forests, etc. Indeed, any prediction model working on standard tabular data could be in principle applied, since the specificities of the data domain are already captured by the user's features $x_u^{\bar{z_p}}$. In this paper, through preliminary experiments, we decided to adopt a random forest method, since it yielded the best and most stable results. The case study in Section V are based on that model.

TABLE I

TRAJECTORY-BASED FEATURES. ALL FEATURES ARE COMPUTED ON THE WHOLE PERIOD, PER DAY-OF-WEEK, AND TIME OF THE DAY (MORNING, AFTERNOON, ETC.). EACH INDICATOR IS AGGREGATED THROUGH COUNT, SUM, MEAN AND STANDARD DEVIATION.

Name	Description
nbr traj	number of trajectories
length	trajectory length in km
duration	trajectory duration in sec
speed	trajectory speed in km/sec

TABLE II

EVENT-BASED FEATURES. ALL FEATURES ARE COMPUTED ON THE WHOLE PERIOD, PER DAY-OF-WEEK, AND TIME OF THE DAY (MORNING, AFTERNOON, ETC.). EACH INDICATOR IS AGGREGATED THROUGH COUNT, SUM, MEAN, AND STANDARD DEVIATION. IN ADDITION, EACH FEATURE IS CALCULATED IN TOTAL AND DIVIDED BY TYPE OF EVENT: HARSH ACCELERATIONS, HARSH BRAKING, HARSH CORNERING, MULTIPLE CORNERING, STARTS, AND STOPS.

Name	Description
nbr events	number of events
duration	event duration in sec
avg acc	average acceleration during the event in km/sec
max acc	maximum acceleration during the event in km/sec
angle	angle of the event

Since an additional objective of risk assessment is to find the possible factors that lead to a crash (whatever the nature of each factor, either causal or simply correlated), we adopt two ways to infer the role played by each feature in the classification. The first one comes as a built-in feature of random forest algorithms, namely the *feature importance* score, which says how much important is overall a feature, though not describing if that is a positive or negative factor. The second way exploits recent results in the explainable AI domain, in particular, the SHAP method [18], which assigns the positive/negative impact of each feature on every single prediction, allowing to make both single-user and collective considerations.

The core of this work lies in the user modeling enabling the classification, i.e., translating the raw yet potentially deep mobility information contained in $H_u^{\bar{z}_p}$ into a set of features $\langle f_1, \ldots, f_m \rangle$ able to capture its significant elements, and in particular those useful for crash prediction. In the following, we illustrate the features used to describe the user's mobility.

C. Trajectory-based features

Position-based features. These features characterize a user using classic indicators of the trajectories and aspects describing them, i.e., length, duration, and speed. Each indicator is aggregated through four operators: counts, sums, means, and standard deviations. Moreover, aggregates are computed over several time periods: morning (6am - 12am of all days), afternoon (12am - 6pm), evening (6pm - 10pm), night (10pm - 6am). This leads to all the combinations ⟨indicator, aggregate type, time period⟩, whose list is summarized in Table I.

Event-based features. Several mobility data sources also contain information about events of various types, detected by the device. They are usually related to acceleration and direction, or to events happened within the device (e.g., receiving a call, in case the device is a mobile phone) or onboard (e.g., a maintenance warning or an action the user made on the car dashboard, in case of vehicles). In our approach, we include this kind of auxiliary information, counting their occurrences or aggregating their associated measures where available, e.g., the acceleration magnitude.

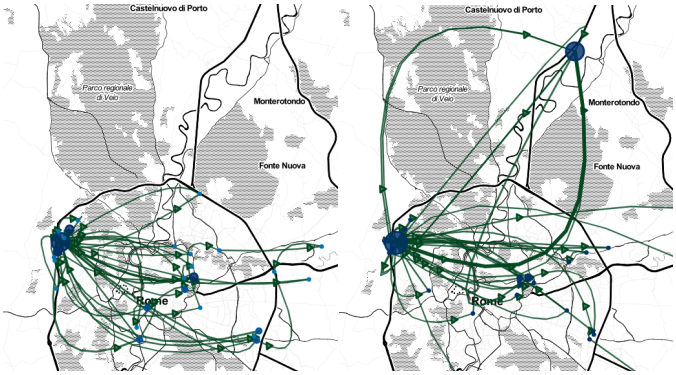

Fig. 1. Temporal evolution of a IMN computed over 2–months periods; changes in mobility are clearly visible.

In this paper we make use of acceleration-based events since they were available in the dataset used for experiments (see Section V-A), yet this can be extended or reduced depending on the data source at hand. Currently, we consider the following events: harsh accelerations, harsh braking, harsh cornering, multiple cornering, vehicle switch-on (start), and switch-off (stop). For each event, it is available the acceleration magnitude (an average computed by the device), maximum acceleration, angle, and duration. The complete list of features is summarized in Table II.

D. IMN-based Mobility features

The Individual Mobility Networks introduced in Section III provide a higher level of aggregation of the user's mobility, also highlighting regular locations and trips. This useful structure is used here to extract three different types of information: (i) the network properties of the IMN, (ii) mobility aggregates focused on high-frequency locations and movements, and (iii) measures of stability in time of the IMN.

Network properties. Network topological measures [26] reveal general properties of the structure of an IMN. For instance, we included the network diameter, helping to understand if the network is compact or it has peripheral nodes very indirectly connected to the others; centrality of the most frequent locations (e.g., home and work), showing how much the mobility converges around them; and the clustering coefficient, describing the tight connectivity of the network.

Mobility information on the IMN. This category includes the geographical dispersion of the mobility (radius of gyration), and mobility statistics on nodes and along edges, such as the number of events that happen on a frequent movement (daily routine trips) or while reaching a frequent location (home, work, etc.). These provide a focused perspective of the general information computed for trajectory-based features.

TABLE III
IMN FEATURES, DIVIDED BY TYPE. STARRED FEATURES (*) WERE COMPUTED BOTH ON ALL THE NETWORK AND FOCUSED ON FREQUENT LOCATIONS. l_i AND l_j ARE THE MOST FREQUENT LOCATIONS l_1, l_2, l_3.

Network features	
nbr locations	number of locations (nodes)*
nbr movements	number of momvements (edges)*
avg degree	average degree, indegree, outdegree
density	graph density
triangles	number of triangles in graph
clus coef	graph clustering coefficient
diameter	graph diameter
eccentricity	graph eccentricity
assortativity	graph assortativity
l_i count	location i count
l_i degree	location i degree
l_i centrality	location i centrality
l_i, l_j count	movement i-j count
l_i, l_j betweennes	movement i-j betweennes
Movement and Events on IMN	
radius	radius of gyration*
movement stats	avg/std movement length/duration
l_i events	location i events
l_i, l_j events	movement i-j events
IMN evolution	
Δ locations	number of locations variation *
Δ radius	radius of gyration variation *
loc jaccard	Jaccard coefficient between locations
loc cosine	Cosine similarity between locations
mov jaccard	Jaccard coefficient between movements
mov cosine	Cosine similarity between movements

IMN evolution. Given two distinct time intervals $\bar{z_1}$ and $\bar{z_2}$ of the same duration, we can expect that some characteristics of the user's mobility remain invariant in the two periods, while others might change. This can provide a perspective on what (and how much) the mobility of the user is stable. A real example is provided in Figure 1, where the IMN of a user is computed on two consecutive 2-month periods and shown over a map. It can be seen that some locations and frequencies (represented by node sizes) remain stable, while others appear/disappear (e.g., the novel North-East frequent location) or change of frequency. The measures of temporal change of the IMN that we adopted focus on general properties of the network, namely the number of locations and the radius of gyration, and the composition and frequency of the locations. The time intervals $\bar{z_1}$ and $\bar{z_2}$ used in the experiments (Section V) have a size of one month, $\bar{z_1}$ covers the earliest part of the historical data used to compute features ($\bar{z_p}$) and $\bar{z_2}$ covers the most recent one, yet different settings are possible.

E. Capturing the Mobility Context

It is intuitively clear that the risk of crash might depend on the context where the user drives. For instance, traversing areas with chaotic traffic is expected to increase the risk of accidents. In addition, driving habits that might abstractly make her a risky driver, such as showing high acceleration rates and high speed, could actually be considered normal if they happen in areas of the city where that is the common behaviour.

TABLE IV
COLLECTIVE AND CONTEXTUAL FEATURES. ALL FEATURES ARE COMPUTED BOTH OVER ALL LOCATIONS OF THE USER AND DIVIDING FREQUENT ONES FROM OCCASIONAL ONES.

Name	Description
nbr traj start	number of trajectory starting
nbr traj stop	number of trajectory stopping
nbr movements	number of trajectory passing by
avg speed	average trajectory speed
nbr crashes	number of crashes
nbr events	number of events
avg acc	average acceleration during the event
max acc	maximum acceleration during the event
avg speed	average speed during the event

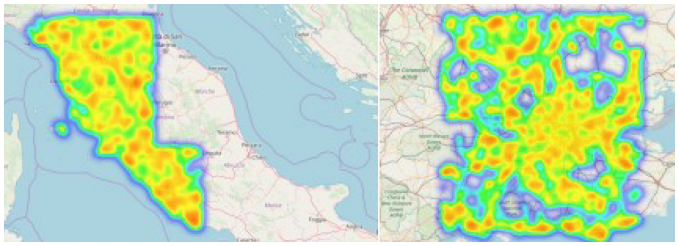

Fig. 2. Geographical areas covered by the data under investigation: Tuscany and Rome in Italy (left), and London in UK (right).

The geospatial context information mentioned above is very difficult to find in existing data sources. For this reason, we computed some estimate contextual indicators directly from the mobility data, by extracting collective aggregates from the history of all users in the dataset. The process starts by defining a spatial partitioning of the geographical area into small sections. This has been performed through a recursive *quadtree* division driven by a dataset of Points-of-Interest[1] (PoI), obtaining finer-resolution partitions in areas with many PoIs (deemed to be more hot and thus more interesting) and coarser ones where they are scattered, e.g., in rural areas. Alternative approaches as regular grids or using administrative boundaries are possible. The second step consists in associating to each geographical section all the data points it contains, computing several aggregates on top of them that characterize the section. In particular, we computed the number of events (including stops and starts), average speed, and acceleration statistics. In the last step, for each user, we consider the geographical sections she stopped in at least once, and compute an average of each characteristic of the sections. The same process is repeated considering only the frequent locations. The complete list is in Table IV.

V. EXPERIMENTS

In this section we present a case study on a dataset of private cars in which we employ the proposed methodology[2]. We first introduce the dataset, and then summarize the results obtained on the crash prediction problem, with a comparison

[1]In our experiments, the OpenStreetMap database was used.

[2]The source code is available at: LinkAvailableAtCameraReady. The dataset is not publicly available.

between our solution and some baselines. Finally, we analyze the predictions returned by the model, trying to infer general useful hints for improving personal driving behaviors.

A. Dataset Description

The dataset considered in our experiments were provided by OctoTelematics[3], and covers three geographical areas (see Figure 2), representing three different situations to be considered in the analyses: a very large city (London, UK), a moderately large city (Rome, Italy), and a whole region, composed of variable-size cities (Tuscany, Italy).

The raw mobility data consists of anonymized GPS traces of vehicles of car insurance customers, containing the following information: *(i)* a list of GPS timestamped *positions* (latitude and longitude); *(ii)* a list of *events* in the form of timestamped position data enriched with labels describing events such as harsh acceleration, harsh braking and (possibly multiple) harsh cornering, with additional accelerometer metrics related to each event position. These data are collected whenever the accelerometer detects an acceleration exceeding predefined parameters; *(iii)* a list of *crashes* in form of timestamped position data related to crash events. Such events were originally detected through algorithms and later filtered by a human operator. The dataset is collected at an average rate of one position every 1.5 minutes, though there are some exceptions.

B. Experimental Setting

Time-wise, in our experiments we consider different time periods, corresponding to prediction times $\tau_p^1 =$ *end of March*, $\ldots, \tau_p^9 =$ *end of November*. The corresponding experiment periods $\bar{z_i}$ are obtained by fixing the history depth τ_h to 3 months (used to compute features) and prediction span to 1 month (the period where crashes are checked). Instead, geographically speaking, we have the three different areas $r \in \{Rome, London, Tuscany\}$, and we analyze the data of about 5000 drivers from each of them. We run the experiments in three different experimental settings, depending on how we consider the temporal and geographical components. In the first setting (*S1*) we keep separated each experiment period $\bar{z_i}$ and each spatial region r from all the others. In particular, for each given pair $(\bar{z_i}, r)$ we train a classification model over the corresponding data of all the users in r, namely $X^{\overline{z_{i,p}}}$ and $y^{\overline{z_{i,p}}}$, and then use the model to make predictions one month later, i.e., it is applied over $X^{\overline{z_{i+1,p}}}$ and the results are compared against the ground truth in $y^{\overline{z_{i+1,p}}}$. Notice that we must have $i + 1 \leq 9$, therefore we obtain a total of $|\{\tau_p^i\}| \times |\{r\}| = 24$ sets of experimental results. In the second setting (*S2*), we still keep regions separated, while all experiment periods are considered together. Users are split into a training set and a test set, following a hold-out division[4], all the 9 experiment periods of a user in the training set are used (as 9 separate records) in the model training and, similarly, all the 9 experiment periods of a user in the test set are used for the model testing. Notice that, while in *S1* we check if we can predict the crash of observed users in the near future using a limited amount of data, in *S2* we try to predict the crash of unobserved users using a consistent amount of data but without a temporal reference. Finally, the third setting (*S3*) amplifies the effects obtained by *S2* by putting the users of different areas in a unique training dataset.

[3]www.octotelematics.com

[4]Cross-validation was also tested, yet results remains basically the same.

C. Dataset Preparation

Before training the classifiers, we face two problems with the datasets analyzed in the various settings. The first one is a class imbalance issue. Indeed there is a very low number of crashes compared to the number of no crashes with an average number of crashes of 3.12% in Tuscany, 1.08% in London, and 2.82% in Rome. We tackled this problem by adopting the SMOTE oversampling approach [27]. The minority class is over-sampled by taking minority class samples and introducing synthetic examples along the line joining the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. We adopt $k = 5$ by default as suggested in [27]. The effect of adopting SMOTE is to improve class balance and to reinforce the presence of the minority class in the decision regions where it appears. We highlight that we re-balance only the training datasets and not the test ones making the evaluation *harder* but more realistic. The second problem is the high dimensionality of the datasets analyzed in various settings. Indeed, the rich data engineering described in the previous sections leads to the construction of more the 400 features, some of them being highly correlated and redundant. This high dimensionality can cause difficulties in the learning of classification models. Thus, we adopt a dimensionality reduction technique based on correlation analysis. We calculated the Pearson correlation coefficient [28] between every pair of features for the various settings. Then, we removed one attribute for each couple having a correlation higher than 0.85. In our experiments, this operation reduced the dimensionality to 162 features, with a balanced presence of trajectory-based, event-based, IMN-based, and contextual features.

D. Evaluation Measures

The objective of crash prediction is to highlight future risky and potentially harmful events, also with the aim of raising an alarm that might help to prevent them. From this perspective, false positives are less critical than false negatives. To this aim we use as main evaluation guidelines [28] the *recall* of the positive class (rec_1), i.e., aiming to find as many risky drivers as possible, and the *precision* of the negative class (pre_0), i.e., aiming to raise no alarm only if we are confident the user is not risky. We account for both aspects considering a weighted f1-measure, i.e., the harmonic mean of precision and recall of the positive class weighted with respect to number of crashes ($f1_1$), and the *area under the roc curve* (*auc*) of the positive class that is the area under the curve comparing the false positive rate (*FPR*) and true positive rate (*TPR*). All measures range from 0 to 1, 1 being the optimum.

TABLE V
ROME: AGGREGATED MEASURES OF PERFORMANCE IN TERMS OF MEANS AND STANDARD DEVIATION OVER DIFFERENT PERIODS FOR S1.

Model	pre_0	rec_1	$f1_1$	*auc*	*crash %*
CST	.000 ± .00	1.000 ± .00	.024 ± .01	.500 ± .00	1.000 ± .00
RFI	.847 ± .36	.877 ± .10	.149 ± .08	.588 ± .05	.848 ± .11
RFP	.704 ± .46	.891 ± .08	.140 ± .07	.574 ± .03	.860 ± .10
RND	.975 ± .02	.486 ± .05	.352 ± .02	.500 ± .00	.502 ± .01

TABLE VI
TUSCANY: AGGREGATED MEASURES OF PERFORMANCE IN TERMS OF MEANS AND STANDARD DEVIATION OVER DIFFERENT PERIODS FOR S1.

Model	pre_0	rec_1	$f1_1$	*auc*	*crash %*
CST	.000 ± .00	1.000 ± .00	.025 ± .01	.500 ± .00	1.000 ± .00
RFI	.702 ± .48	.992 ± .01	.056 ± .04	.719 ± .05	.968 ± .04
RFP	.425 ± .53	.992 ± .01	.042 ± .03	.577 ± .04	.983 ± .03
RND	.973 ± .02	.488 ± .03	.355 ± .01	.500 ± .00	.498 ± .01

E. Crash Prediction Evaluation

We experimented with different classifiers to account for performances, yet also considering their interpretability. Simple and partially interpretable decision tree and k-NN classifiers fail in reaching acceptable performances, while *Random forests* (*RF*) performances overcome those of multi-layer perceptron algorithms, making RF the best candidate. In the following, we report the results obtained using RF classifiers with 100 estimators, i.e., 100 trees in the forest, allowing leaves with at least 1% of the training data, and with a cost matrix weighting a crash 100 times more than a no crash. We show the effectiveness of our approach by comparing against three alternative approaches. The first two are simple baselines: a *constant* classifier (CST) always returning the relevant class (crash); a *random* classifier (RND), predicting uniformly ad random crash or not crash. The third one, instead, adopts a RF using only features from the state-of-the-art literature of crash prediction (RFP), such as average speed, number of trajectories, number of breaks, etc. Our proposed classifier (RFI) improves over RFP by extending the classical features used in literature with the much more sophisticated IMN-based and contextual features described in Section IV-B.

Tables V, VI, VII report the result for S1, showing the evaluation measures previously classifiers and the percentage of crashes returned by the classifiers for Rome, Tuscany, and London areas, respectively, averaged among the various periods. Table VIII reports the same indicators for the experimental settings S2 (first three rows) and S3 (last row). The overall results we observe in the various settings and tables are the following. The simultaneous analysis of the reported indicators shows that RFI provides the best and most reliable performances. Indeed, the CST baseline obviously has the highest recall but a zero precision on no crashes, making it useless for practical usage. On the other hand, RND easily gets a high precision of no crashes, thanks to the high imbalance of data, but it loses half of the effective crashes with a recall of less than 0.5. RFP gives a better trade-off between precision and recall than CST and RND, but shows a very high number of records returned as crashes and an *auc* just slightly better than CST and RND, with a value around 0.6. On the other hand, RFI always has similar or larger levels of precision and recalls of RFP, and has systematically a higher *auc*, also labeling as crashes a number of records consistently lower than RFP.

TABLE VII
LONDON: AGGREGATED MEASURES OF PERFORMANCE IN TERMS OF MEANS AND STANDARD DEVIATION OVER DIFFERENT PERIODS FOR S1.

model	pre_0	rec_1	$f1_1$	*auc*	*crash %*
CST	.000 ± .00	1.000 ± .00	.009 ± .00	.500 ± .00	1.00 ± .00
RFI	1.000 ± .00	.994 ± .01	.574 ± .02	.962 ± .01	.086 ± .01
RFP	.994 ± .00	.719 ± .10	.308 ± .05	.612 ± .04	.572 ± .10
RND	.991 ± .00	.499 ± .08	.341 ± .00	.500 ± .00	.501 ± .01

TABLE VIII
PERFORMANCE FOR DIFFERENT AREAS FOR S2 (TOP) AND S3 (BOTTOM).

area	*model*	pre_0	rec_1	$f1_1$	*auc*	*crash %*
London	CST	.000	1.00	.010	.500	1.00
	RFI	1.00	1.00	.580	.955	.087
	RFP	.992	.624	.329	.574	.533
	RND	.990	.489	.344	.500	.495
Rome	CST	.000	1.00	.028	.500	1.00
	RFI	.985	.882	.216	.619	.776
	RFP	.978	.866	.180	.586	.822
	RND	.972	.500	.361	.500	.493
Tuscany	CST	.000	1.00	.029	.500	1.00
	RFI	.993	.944	.243	.775	.745
	RFP	.973	.970	.061	.584	.967
	RND	.969	.480	.355	.500	.504
All	CST	.000	1.00	.022	.500	1.00
	RFI	.999	.991	.206	.776	.787
	RFP	.975	.996	.025	.641	.997
	RND	.977	.485	.352	.500	.498

In the experimental setting S1 we observe different behaviors of RFI in the three different areas considered, and reported in Tables V, VI, and VII. In London, the RFI classifier labels as crashes only the 8% of the records in the test sets against the 84% and 97% in Rome and Tuscany. However, it has at the same time the highest pre_0, rec_1, $f1_1$, and *auc*. Notice that the other methods considered show much worse results. In other words, the new features introduced in this paper appear to make crashes easy to predict in London. Understanding the reasons for this effect is part of our future works.

The results for S2 and S3 are reported in Table VIII. We observe how the increased number of available records for the training leads to a not negligible improvement in the performance of the classifiers for S2 in the Rome, Tuscany, and London areas when compared to those in Tables V, VI, and VII. In addition, the setting S3 that puts together records from all the different areas (All rows in Table VIII) leads to a classifier even better than those resulting from S2. We highlight in Figure 3 the Receiver Operating Characteristic (ROC) curve of the classifiers for the experimental setting S2 for Rome, Tuscany, and London, and S3 for all the data records. These plots show the evidence that London classifiers are much more accurate than the others and that RFI classifiers markedly benefit from the usage of IMN-based and contextual features with respect to RFP, whose ROC curve is always below.

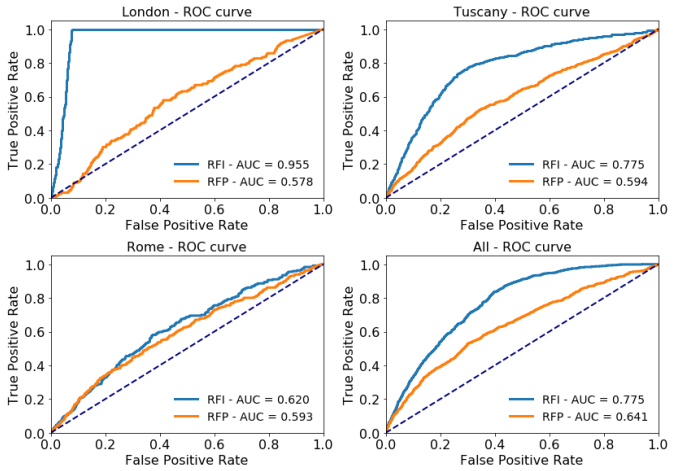

Fig. 3. Receiver Operating Characteristic (ROC) curve for different areas for S2 (London, Rome, Tuscany) and S3 (All).

TABLE IX
MODELS TRANSFERABILITY IN TERMS OF PERFORMANCE OF A CLASSIFIER IN AN AREA DIFFERENT FROM THE ONE USED FOR LEARNING.

model	*area*	pre_0	rec_1	$f1_1$	*auc*	*crash %*
London	Rome	.971	.000	.493	.500	.000
	Tuscany	.970	.000	.492	.501	.000
Rome	London	.996	.722	.431	.772	.309
	Tuscany	.977	.692	.317	.561	.601
Tuscany	London	.997	.767	.473	.864	.217
	Rome	.976	.915	.120	.566	.899
All	London	.999	.977	.295	.874	.615
	Rome	.993	.980	.109	.615	.915
	Tuscany	.989	.886	.269	.770	.701

F. Model Transferability

We stress the classifiers of S2 showing which are their performance when applied on data records of areas not used for learning. Similar results are obtained for S1, not reported here for better readability. Table IX reports the performance of the classifier trained on London when applied to the Rome and Tuscany test sets, the performance of the classifier trained on Rome when applied to London and Tuscany test sets, etc. We observe that the London classifier is not able to label any driver as possible crash, showing that it captures very local (and locally very effective, as seen in the previous section) aspects of the London crash patterns. Moreover, both the Rome and Tuscany classifiers succeed in finding crashes in London even though with a recall lower than the one of the classifiers trained specifically on it. Hence, it seems that learning to detect crashes in London is easier than other areas (at least with IMN and context features). On the other hand, the Tuscany and Rome classifiers are quite interchangeable having a high transferability between them. The bottom row of the table shows that a classifier trained on all the datasets in S3 has similar performances over all areas, suggesting that the area of learning is the main factor affecting the model behavior.

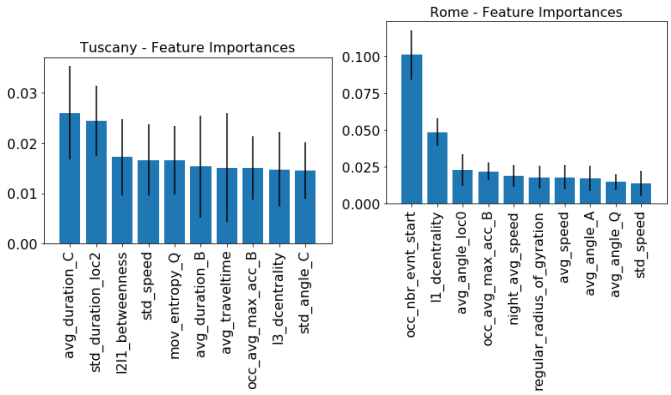

Fig. 4. Feature importance of the classifiers for two areas of S2 (Tuscany and Rome). Plots for Rome and the whole dataset are omitted for space limitation.

G. Risk Assessment Analysis

Besides building a reliable car crash predictor in terms of performances, the objectives of this work include the risk assessment analysis aimed to understand which behaviors in a driver more likely could lead to future crashes. We can accomplish this task by extracting from the trained classifiers the knowledge describing different driving profiles associated with the users that are more prone to have future crashes.

Since we are adopting RF classifiers, we can easily extract the feature importance computed as the (normalized) total reduction of the error brought by a certain feature (also known as the Gini importance [28]) for each decision tree in the forest, averaging among the values obtained from each tree. We report examples of these values for S2 Figure 4. These plots provide a first idea of which are the most discriminating features, yet not helping in understanding the specific role played by each feature or the reasons for the model decisions.

To overcome these limitations, we adopted methods brought from the "explainable AI" field [3], in particular, the SHapley Additive exPlanations (SHAP) method [18]. SHAP connects game theory with local explanations based on feature importance. In particular, it exploits the Shapely values of a conditional expectation function of the classifier to explain (in our case, the crash predictor), providing the local unique additive feature importance for each specific record. The higher is a shapely value, the higher is the contribution of the feature. If the shapely value is positive, it contributes towards the positive class (crash). Otherwise, it contributes towards the negative class (no crash). In Figure 5, we report the shapely values for the records in the test set classified as crash by the classifier learned in S3, using the *force plots* introduced in [29]. Every colored line represents a feature, each horizontal position is a different user, and vertical values show the feature contribution to the classification. Features pushing the prediction towards crash are shown in red, those pushing the prediction towards no crash are in blue. The purpose of this Figure is to show that there are clusters of similar records classified as crashes for different reasons.

We retrieve these different clusters (and thus the different

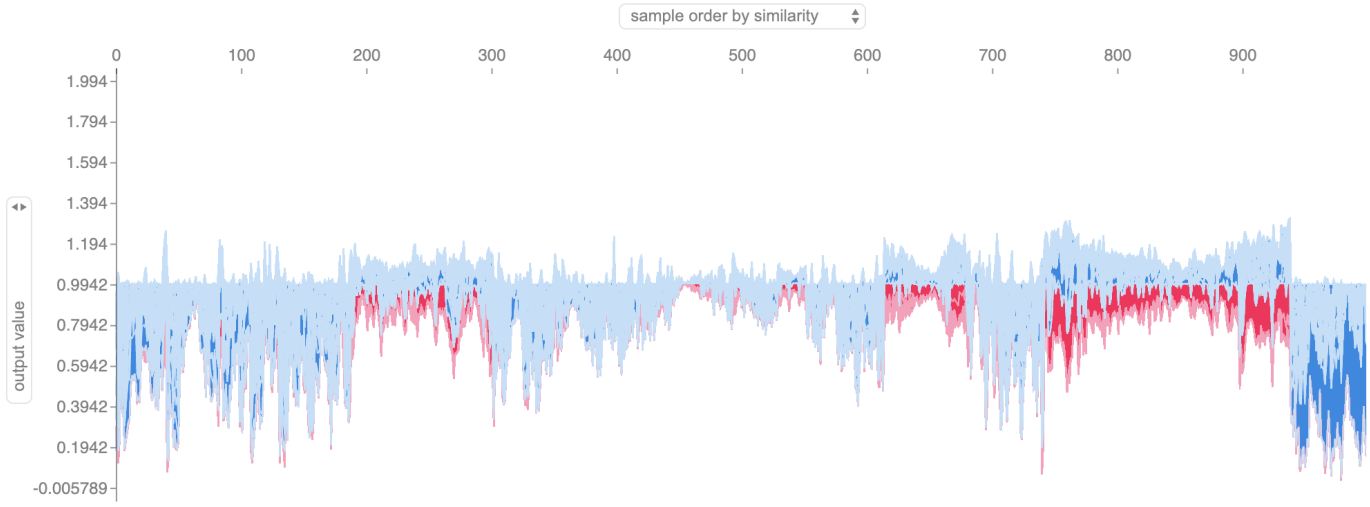

Fig. 5. Shap values of the records in the test set for the classifier learnt in S3. Every colored line represents a feature, each horizontal position is a different user, and vertical values show the feature contribution to the classification. Features pushing the prediction towards crash are shown in red, those pushing the prediction towards no crash are in blue. (*Force plots* introduced in [29])

reasons they represent) by adopting the following procedure we propose for risk assessment analysis. Given the shapely values computed for a set of records, we cluster them using a centroid-based approach. We adopted K-Means and observing the Sum of Squared Error (SSE) distribution we selected $k = 10$ as the number of clusters. Then, for each cluster, we select a *medoid*, i.e., an element representing the cluster that minimizes the distance between all the other records in the cluster in terms of shapely values. Finally, we report the shapely values of these prototypes as profiles of users with a probable crash with an indication of the reasons for the crashes. Due to space constraints we only report in Figures 6 and 7 three prototypes for S2 Tuscany and S3 respectively.

An analysis of the prototypes confirms that IMN-based features and collective features are fundamental for detecting crashes. Indeed, the average maximum acceleration of break events in areas visited occasionally performed by other users is crucial in pushing towards crash in Tuscany for the three prototypes reported. With respect to the classifier returned by S3, the prototypes reveal that a high Jaccard of movements among different IMNs (i.e., high temporal stability in movement routines) pushes the decision towards crash. Another feature having this effect is the number of acceleration and break events between the second and third most visited locations.

VI. Conclusion

In this paper we introduced and tackled the (long-term) car crash prediction problem and its associated task of risk assessment. The solution proposed consists in extracting sophisticated features of the user that capture not only basic characteristics of her mobility, but also higher-level information derived from a network view of her mobility history as well as contextual knowledge directly inferred through analysis of the collective data of all users. On top of such features, many standard machine learning models can be used, among which Random Forests proved to be the most promising for this application. Experiments on real data showed that our solution outperforms basic solutions based on state-of-art features, and a preliminary inspection of the prediction models through explainable AI methods allowed us to identify a few representative features associated with crash risk.

Ongoing and future works on this line of research include an extension of all the steps of the proposed solution. The IMN representation could be refined by annotating trips and locations with their purpose [12], by recognizing driving *moods* (e.g., through unsupervised analysis of speeds and accelerations, or driving through dangerous intersections [20]), or by better describing the evolution of driving habits. Contextual data might be expanded, e.g., by including external information, such as the presence of POIs. Adaptation strategies should be designed in order to make strong models built in one area transferable to other places. Finally, the risk assessment could be developed to better explain the predictions (e.g., using rule-based methods [30]) and infer actionable changes in driving habits that might help the user exit the risk zone.

Acknowledgment

This work is partially supported by the European Community H2020 programme under the funding scheme *Track &Know* (Big Data for Mobility Tracking Knowledge Extraction in Urban Areas), G.A. 780754, https://trackandknowproject.eu/. We thank Sistematica for the dataset.

References

[1] L. Longhi and M. Nanni, "Car telematics big data analytics for insurance and innovative mobility services," *JAIHC*, pp. 1–11, 2019.
[2] Y. Wang *et al.*, "Machine learning methods for driving risk prediction," in *SIGSPATIAL Workshop*. ACM, 2017, p. 10.
[3] R. Guidotti, A. Monreale, S. Ruggieri *et al.*, "A survey of methods for explaining black box models," *ACM CSUR*, vol. 51, no. 5, p. 93, 2019.
[4] C. Lee *et al.*, "Real-time crash prediction model for application to crash prevention in freeway traffic," *TRR*, vol. 1840, no. 1, pp. 67–77, 2003.

Fig. 6. Shap values of three prototypes for the classifier learnt in S2 Tuscany.

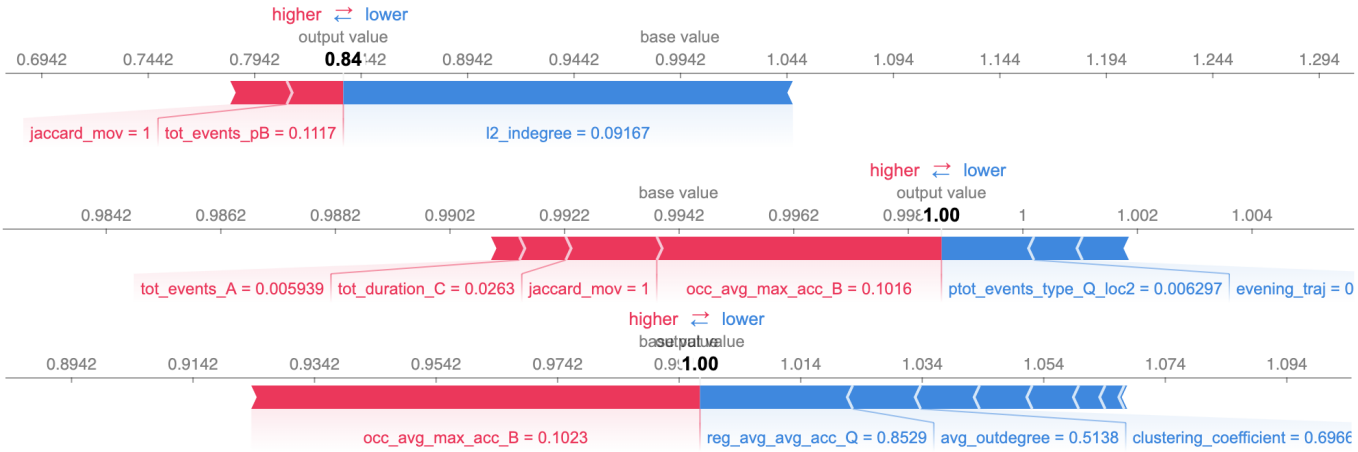

Fig. 7. Shap values of three prototypes for the classifier learnt in S3.

[5] Y. Ba *et al.*, "Crash prediction with behavioral and physiological features for advanced vehicle collision avoidance system," *TR-C*, 2017.

[6] L. A. Cruz, K. Zeitouni, and J. A. F. de Macedo, "Trajectory prediction from a mass of sparse and missing external sensor data," in *MDM*. IEEE, 2019, pp. 310–319.

[7] R. Trasarti, F. Pinelli, M. Nanni, and F. Giannotti, "Mining mobility user profiles for car pooling," in *SIGKDD*. ACM, 2011, pp. 1190–1198.

[8] D. Pedreschi *et al.*, "Meaningful explanations of black box ai decision systems," in *AAAI*, vol. 33, 2019, pp. 9780–9784.

[9] L. Pappalardo *et al.*, "Returners and explorers dichotomy in human mobility," *Nature communications*, vol. 6, p. 8166, 2015.

[10] R. Guidotti *et al.*, "Tosca: two-steps clustering algorithm for personal locations detection," in *SIGSPATIAL*. ACM, 2015, p. 38.

[11] R. Guidotti, M. Nanni, and F. Sbolgi, "Data-driven location annotation for fleet mobility modeling," in *BMDA*, 2019.

[12] S. Rinzivillo *et al.*, "The purpose of motion: Learning activities from individual mobility networks," in *DSAA*. IEEE, 2014, pp. 312–318.

[13] R. Guidotti *et al.*, "There's a path for everyone: A data-driven personal model reproducing mobility agendas," in *DSAA*. IEEE, 2017.

[14] K. Zeitouni *et al.*, "Spatial decision tree-application to traffic risk analysis," in *ICCSA*. IEEE, 2001, pp. 203–207.

[15] A. A. Freitas, "Comprehensible classification models: a position paper," *ACM SIGKDD explorations newsletter*, vol. 15, no. 1, pp. 1–10, 2014.

[16] A. Adadi *et al.*, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE*, vol. 6, pp. 52 138–52 160, 2018.

[17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE TKDE*, vol. 22, no. 10, pp. 1345–1359, 2009.

[18] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NIPS*, 2017, pp. 4765–4774.

[19] J. Wang, W. Xu, and Y. Gong, "Real-time driving danger level prediction," Nov. 23 2010, uS Patent 7,839,292.

[20] F. D. Salim *et al.*, "Collision pattern modeling and real-time collision detection at road intersections," in *ITSC*. IEEE, 2007, pp. 161–166.

[21] M. A. Abdel-Aty and R. Pemmanaboina, "Calibrating a real-time traffic crash-prediction model using archived weather and its traffic data," *IEEE TITS*, vol. 7, no. 2, pp. 167–174, 2006.

[22] F. L. Mannering *et al.*, "Analytic methods in accident research: Methodological frontier and future directions," *AMAR*, vol. 1, pp. 1–22, 2014.

[23] Y.-J. Kweon *et al.*, "Development of crash prediction models with individual vehicular data," *TR-C*, vol. 19, no. 6, pp. 1353–1363, 2011.

[24] D. Lord and F. Mannering, "The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives," *TR-A*, vol. 44, no. 5, pp. 291–305, 2010.

[25] R. Trasarti *et al.*, "Myway: Location prediction via mobility profiling," *Information Systems*, vol. 64, pp. 350–367, 2017.

[26] G. Rossetti *et al.*, "A supervised approach for intra-/inter-community interaction prediction in dynamic social networks," *SNAM*, 2016.

[27] N. V. Chawla *et al.*, "Smote: synthetic minority over-sampling technique," *JAIR*, vol. 16, pp. 321–357, 2002.

[28] P.-N. Tan, *Introduction to data mining*. Pearson Education India, 2018.

[29] S. M. Lundberg *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia," *NBE*, vol. 2, no. 10, p. 749, 2018.

[30] R. Guidotti, A. Monreale *et al.*, "Factual and counterfactual explanations for black box decision making," *IEEE Intelligent Systems*, 2019.