Robust Data-Driven Framework for Driver Behavior Profiling Using Supervised Machine Learning

Abdalla Ebrahim Abdelrahman, *Student Member, IEEE*, Hossam S. Hassanein, *Fellow, IEEE*, and Najah Abu-Ali, *Member, IEEE*

***Abstract*— Driver behavior profiling has been gaining increased attention due to its relevance in many applications. For instance, car insurance telematics and fleet management entities have been recently using smartphones' embedded sensors, On-Board Diagnostics II (OBDII) units and other on-board IoT devices to collect data on vehicles' behavior and evaluate the risk profile of drivers. In this context, this paper presents a robust data-driven framework for calculating drivers' risk profile measured in terms of the additive inverse of the predicted risk probability. The Strategic Highway Research Program 2 (SHRP2) naturalistic driving study (NDS) dataset, which is the largest dataset of its kind to date, is utilized to build the risk prediction models. Crash and near-crash events are used to quantify riskiness whereas balanced baseline driving events (i.e., events captured during normal day to day driving episodes) are used to reflect total exposure or driving time per driver. Thirteen mutually exclusive behavioral risk predictors are identified, and the feature matrix is formulated. A sensitivity analysis is then performed to find the best number of balanced baseline events below which drivers are filtered out. Different machine learning models are selected, customized, and compared to achieve best risk prediction performance. Finally, the utilization of the proposed prediction model within an envisioned driver profiling cloud-based framework is briefly discussed.**

***Index Terms*— Internet of intelligent vehicles (IoIV), driving behavior profiling, data-driven applications, intelligent transportation systems (ITS), prediction models, vehicle-to-cloud (V2C) applications.**

I. Introduction

THE Internet of Things (IoT) is gaining increasing relevance in many applications due the recent advancements in communications, identification and sensing technology [1], [2]. IoT enables objects to sense and communicate information in real-time which facilitates information exchange, analysis and decision making [3]. According to Gartner report in [4], it is expected that 20 billion IoT devices will be connected by 2020. This new wave of technology has gained its significance in a wide range of applications such as in smart homes [5], [6], connected wearables [7], and intelligent transportation systems' (ITS) applications [8] including driver risk profiling, which is the focus of this paper.

According to the World Health Organization (WHO) global status report in road safety, it is anticipated that road crashes will be the seventh leading cause of death in 2030 unless serious actions are taken [9]. Recently, researchers have been utilizing the Internet of Intelligent Vehicles (IoIV) technology, with attention on ensuring safe driving [10]. IoIV technology refers to the dynamic mobile communication between vehicles (V2V), vehicles and road infrastructures (V2I), vehicles and humans (V2H) or vehicles and cloud (V2C) with the primary objective of minimizing driving risk and ensuring a better driving experience.

Driver risk profiling is an emerging V2C driving application which has particular significance in the fleet management and car insurance telematics domains [11]. In fleet management, fleet administrators are keen on tracking the behavior of their drivers to ensure the safety of their fleets and the roads. Likewise, car companies are adopting a new insurance paradigm called pay-how-you-drive (PHYD) in which insurance premiums are adapted according to the real-time behavior of drivers. In both domains, data that reflect a subject vehicle's (sv) behavior is collected using smartphones' embedded sensors and/or On-Board Diagnostics II (OBDII) units, and is then sent to the cloud for analysis. In the cloud, different figures of merit (FOMs) are typically calculated for each trip using collected data and a driver's risk score is provided accordingly.

Modeling the actual risk score based on the detected FOMs is viewed by many as an intricate problem. The reason is that the process of designing efficient scoring models necessitates the existence of enough and reliable data, which is not always available. Consequently, different insurance companies have been adopting several scoring models that assign different weights to each FOM [12]. Although several insurers are viewing the number of harsh braking events as the best risk predictor, there is no common agreement about the statistical significance of such measure.

Among the different data collection approaches, naturalistic driving studies (NDSs) have recently prevailed [13]–[15]. NDSs provide researchers with the opportunity to study the

Manuscript received September 16, 2018; revised April 13, 2019, December 4, 2019, March 9, 2020, and June 21, 2020; accepted October 30, 2020. Date of publication November 17, 2020; date of current version March 29, 2022. This work was supported in part by the Research Center RTTSRC-4-2013 provided by the Roadway Transportation and Traffic Safety Research Center, United Arab Emirates University under Project 31R014 and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant STPGP 479248. The Associate Editor for this article was C. Wu. *(Corresponding author: Abdalla Ebrahim Abdelrahman.)*

Abdalla Ebrahim Abdelrahman is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: a.abdelrahman@queensu.ca).

Hossam S. Hassanein is with the School of Computing, Queen's University, Kingston, ON K7L 2N8, Canada (e-mail: hossam@cs.queensu.ca).

Najah Abu-Ali is with the College of Information Technology, United Arab Emirates University, Al-Ain, UAE (e-mail: najah@uaeu.ac.ae).

Digital Object Identifier 10.1109/TITS.2020.3035700

behavior of drivers, explore the different driving patterns, and provide data-driven approaches for calculating the risk associated with several driving behaviors [16]. For instance, the Strategic Highway Research Program 2 (SHRP2) NDS dataset offers an unprecedented amount of driving context data for almost 9,000 recorded crash and near-crash events and more than 20,000 balanced base-line events (i.e., normal driving events proportional to the total driving per driver) for more than 3,000 drivers [17]. The collected data gives not only the opportunity to study the prevalence of behavioral factors during risky events but also their prevalence through normal driving episodes, which enables the conduction of statistically sound studies. This dataset is considered by far the largest of its kind. Consequently, the efficient utilization of such dataset can lead to a formulation of more robust driving risk models and can provide more insights into the significance of each risk predictor.

This paper presents a novel robust data-driven framework for evaluating drivers' risk scores and the incorporation of this framework in a cloud-based driver profiling system.

The main contributions of our paper can be summarized as follows:

1) We provide a practical, robust data-driven framework for calculating drivers' risk profiles (i.e., aggregated risk scores) as a function of the predicted risk probability of their behavioral patterns. This is achieved by the utilization of the behavioral context information during base-line, crash and near-crash events of SHRP2 dataset.
2) A comparative study between selected and customized machine learning algorithms is performed to determine the best performing algorithm for the risk prediction problem. Algorithms are compared in terms of their average performance and their performance consistency through various testing samples.

The remainder of this paper is organized as follows. In section II, a background review and the related work are provided. Section III presents the proposed risk profiling framework. Also, the mathematical formulation of the risk prediction problem is introduced. In section IV, the adopted data filtering and pre-processing processes are discussed. In section V, machine learning algorithms that are utilized to predict riskiness are presented. The selection process of these algorithms and the customization of their hyper-parameters for the presented risk prediction problem is motivated. In section VI, performance assessment metrics that are employed to measure the performance of the utilized models are discussed. Results and discussion are then presented in section VII. An envisioned cloud-based profiling system based on the developed risk prediction model is highlighted in section VIII and conclusions are finally presented in section IX.

II. Background and Related Work

A. Driving Behavior Profiling

Driver profiling is based on acquiring continuous information about the behavior of an *sv* driver through the use of unobtrusive devices such as OBDII units and/or smartphones [18]. This data is then processed and classified into driving

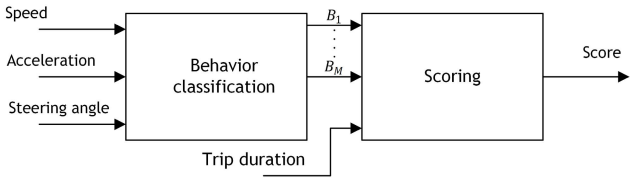


Fig. 1. Driver behavior classification and scoring.

behaviors which are inputted, along with other FOMs such as trip duration, to a scoring function as shown in Figure 1. A scoring function is a model that can take different forms and assigns weights to the FOMs according to their risk impact [12]. Conventionally, there are four driving behavioral FOMs that are utilized as risk quantification measures (i.e., risk predictors) to calculate a risk score for a certain driver [11], [12]. These FOMs are:

1) **Braking**: number of harsh braking events.
2) **Speeding (relative or absolute)**: number of excessive speeding events whether more than the speed limit or relatively higher than surrounding vehicles.
3) **Cornering**: number of events when turning at a higher than posted speed.
4) **Acceleration**: number of hard acceleration events.

Several industrial products and research frameworks have been implemented and proposed. For instance, car insurance companies have developed different smartphone applications that are compatible with IOS and Android operating systems and are capable of detecting and evaluating the behavior of drivers by utilizing smartphones' sensors such as: accelerometers, magnetometers and GPSs. Examples include TDMyAdvantge, Aviva RateMyDrive and StateFarm DriverFeedback applications [19]. The aggregated scores over many trips are used to adjust the drivers' insurance premiums.

Research in this field has taken two main directions:

1) Driver behavior detection and classification. This includes the detection of certain events such as: aggressive acceleration, aggressive lane change, etc. [20]–[24].
2) The development of risk prediction and scoring functions that accurately reflect risk rate given the detected behaviors [25], [26].

While the former contains many contributions; proposals and frameworks, the latter has very few. Indeed, the choice of scoring functions has been very subjective due to the absence of a frame of reference which is due to the lack of large-scale and reliable datasets.

Large-scale driving datasets are necessary to develop a reliable data-driven risk prediction model that can infer the statistical dependence between detected behaviors and the expected driving risk (e.g., crash and near-crash probability). Such a model is crucial to provide drivers with fair risk scores based on the risk potential of their different behavioral patterns. The Developed scoring function can be used within a smartphone application or in a cloud server after the detection/classification of driving behaviors during a specific driving trip.

Driver behavior detection has been extensively studied in the literature. Authors in [27] evaluated the performance of

four machine learning algorithms in detecting seven different driving maneuvers. Authors concluded that Random Forests algorithm is superior over other algorithms in the detection of such events. In [28] the authors proposed "DriveSafe" iPhone application that is capable of detecting drowsy and distracted driving behaviors by utilizing the iPhone's built-in rear-camera, microphone, inertial sensors, and GPS. Authors in [29] utilized DriveSafe application to provide a large-scale naturalistic driving (ND) dataset (UAH-DriveSet) in two road types (i.e., highways and secondary roads). With 500 minutes of publicly available ND data, UAH-DriveSet is expected to facilitate the research in the field of driving behavior detection/classification. In [23], authors proposed an HMM-based model to detect abrupt and normal driving maneuvers in both longitudinal and lateral directions. Events were detected using smartphones and authors claimed to have a classification accuracy of $\sim 95\%$. Authors in [18] proposed an application called MobiDriveScore that acquires data from a smartphone and a vehicle's network (i.e., CAN-bus) to detect risky events. A smartphone application called CarSafe was proposed in [30] to detect dangerous behaviors. Authors utilized smartphones' dual cameras to detect a number of dangerous events. The smartphone was mounted on the dash of the car. They used the front camera to detect drowsiness and distraction whereas the rear camera was utilized to detect tailgating and unintentional lane changing. A fuzzy logic based smartphone application was proposed in [25]. A complete driving behavior detection and scoring system was proposed and discussed and four unique driving events were detected with high accuracy by fusing smartphone's accelerometer, gravity, magnetic, and GPS data. Moreover, authors used two different smartphones with different sampling rates and resolutions and compared their detection performances which were found to be consistent. Similar work was presented in [31] in which authors used accelerometer, gyroscope, and magnetometer sensors of a smartphone to detect sharp turning, aggressive acceleration and abrupt lane changing, and sudden braking. To compensate for the varying time of events, a dynamic time wrapping (DTW) algorithm was implemented and maneuvers were then classified according to their risk level using a binary Bayesian classifier. Other proposals such in [32] aimed to predict the driving behavior at signed intersections using HMM-based model. More recently, authors in [20] proposed five HMM-based models that are capable of classifying the behavior of the driver by taking into account the behavior of surrounding vehicles. Models were trained and tested using the 100-CAR dataset.

Despite the research efforts mentioned above in event detection and driver behavior classification field, contributions in formulating reliable scoring functions are still very primitive [25], which motivated the formulation of reliable data-driven scoring models presented in this work.

B. SHRP2 NDS Dataset

Human error contributes to approximately 90% of crashes [33]. In order to examine the influence of different driving behaviors on the crash rate, different approaches have been proposed including the ND data collection approach [13]. ND data collection methodology provides three important advantages over other data collection methods [14]:

1) Detailed information about the behavior of a driver prior to a crash or near-crash events.
2) Exposure data, which provides vital information about the frequency of occurrence of different driving behaviors during normal driving episodes.
3) The amount and reliability of collected data paves the way for conducting statistically sound studies.

The Virginia Tech Transportation Institute (VTTI) has been pioneering this approach since the beginning of this century with two large-scale data collection projects, the 100-CAR NDS and more recently the SHRP2 NDS. In SHRP2 NDS, 3542 drivers were recruited in six different sites in the United States, and their vehicles were equipped with unobtrusive data acquisition systems (DASs) containing mainly forward radar sensors, video cameras, OBD units to acquire the vehicle's CAN bus information, and global positioning systems (GPSs). Participants were then asked to use their vehicles in their normal day to day driving routine. Data was continuously recorded which resulted in more than 35 million miles of driving data. This is, by far, the largest amount of naturalistic driving data ever recorded to date [14].

Data reductionists were then able to extract almost 9,000 risky events which are comprised of crash and near-crash events. Moreover, normal driving events where randomly chosen for each driver to offer exposure information. These episodes are called balanced baseline events as their number is balanced with the total driving time of a driver. The operational definitions for each type of these events can be found in [34] and are briefly as follows:

1) Crash: any contact made with an object (alive or inanimate) either moving, or fixed, by the vehicle driven by sv driver. This also includes inadvertent departures of the roadway if contact is made.
2) Near crash: any driving maneuver(s) performed by the sv driver that will avoid their vehicle from being involved in a crash.
3) Balanced baseline events: epochs of data selected to provide exposure information. They are 21 seconds long and their number is proportional to the total driving time for each driver.

In this work, we utilized the information of 1836 crashes of all severity levels which represent $\sim 6\%$ of the overall number of events, 6881 near-crash events which constitute $\sim 24\%$ of the overall number of events, and 20179 baseline events which represent $\sim 70\%$ of the overall number of events. The number of baseline events reflect the total driving time of drivers over the period of SHRP2 study. Baseline events were used in this work to provide a "*snapshot* "of the behavioral patterns of drivers on the long-term. The detailed selection criteria of the number of baseline events per driver can be found in [34]. The dominant driving behaviors prior to crash/near-crash events or during baseline events were extracted and recorded from the collected SHRP2 data by VTTI data reductionists.

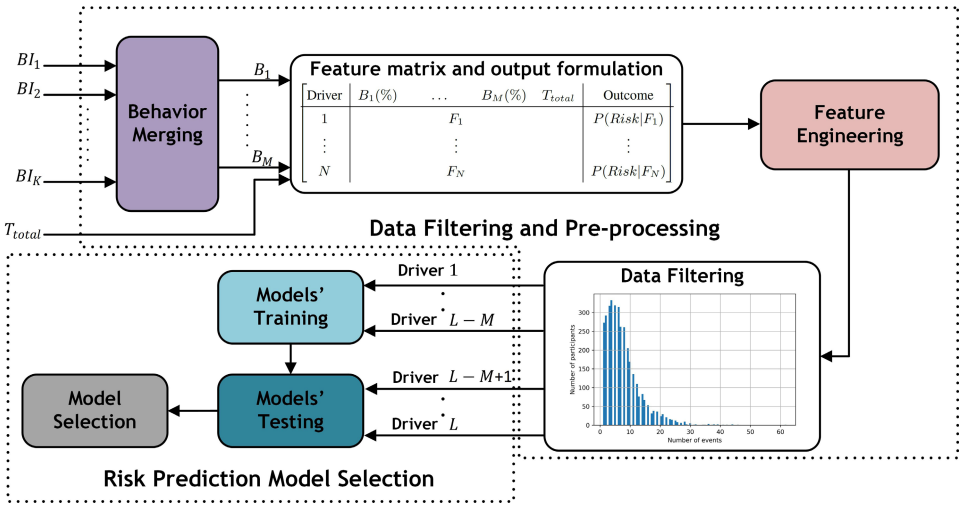


Fig. 2. Block diagram of the adopted data filtering and pre-processing on SHRP2 raw data.

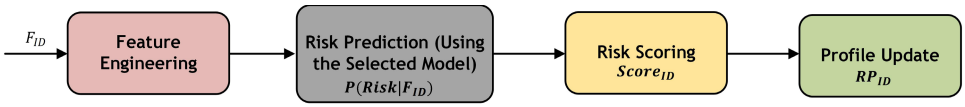


Fig. 3. Block diagram of the proposed driver's risk profiling system.

III. Proposed Driver Profiling Framework

In this section, the mathematical formulation of the proposed driver risk profiling framework is presented. Figure 2 depicts the block diagram of the adopted offline data filtering, pre-processing, and risk prediction model selection processes. The figure shows the logical sequence of processes applied on the SHRP2 raw data towards a robust risk prediction for different behavioral patterns. Data filtering and pre-processing process consists of merging the raw SHRP2 contextual driving behaviors to increase their importances, feature and output engineering, and filtering out unrepresentative data, whereas the risk prediction model selection phase is composed of the training, testing, and selection of the risk prediction models. Figure 3 shows the online risk profiling process which is composed of the online risk prediction, driver scoring and profiling. The specifics of the system's individual components are explained in sections IV and V.

In the proposed framework, the long-term predicted risk of different behavioral patterns are used to reflect the short-term per trip risk scores. To predict the long-term driving risk, each driver is represented by a feature vector denoted by "F_{ID}" which is expressed as:

$$F_{ID} = \begin{bmatrix} B_1(\%) & \dots & B_M(\%) & T_{total} \end{bmatrix} \tag{1}$$

where the vector entries "$B_i(\%)$" represent the frequency of occurrence of each identified behavior with respect to other behaviors and T_{total} is a categorical variable that reflects the total exposure (driving) time for a driver represented here in terms of the total number of base-line driving events. In this work, thirteen mutually exclusive driving behaviors have been identified as risk predictors as will be detailed in the following section. The identified behaviors are depicted in table I with a brief description of each. The risk prediction is formulated as both a classification and a regression problem as will be discussed in the following section. The risk prediction is initially defined as the process:

$$\mathcal{F} : F_{ID} \rightarrow P(Risk|F_{ID}) \tag{2}$$

where $P(Risk|F_{ID})$ is the probability of driver ID being involved in a risky event given his/her feature vector F_{ID}. $P(Risk|F_{ID})$ is governed by the summation of the crash (C) and near-crash (NC) conditional probabilites as shown in equation 3:

$$P(Risk|F_{ID}) = P(C|F_{ID}) + P(NC|F_{ID}) \tag{3}$$

These conditional probabilities are expressed herein in terms of crash, near-crash, and captured baseline events for each driver as follows:

$$P(C|F_{ID}) = \frac{NC_{ID}}{NT_{ID}} \tag{4}$$

$$P(NC|F_{ID}) = \frac{NNC_{ID}}{NT_{ID}} \tag{5}$$

where NC_{ID} and NNC_{ID} are respectively the numbers of recorded crash and near crash events for driver ID, and NT_{ID} represents the total number of recorded events for driver ID. A driver's score is then computed in terms of the additive inverse of $P(Risk|F_{ID})$ as shown in equation 6.

$$Score_{ID} = 1 - P(Risk|F_{ID}) \tag{6}$$

Practically, scores are calculated for each trip. In this context, a one-to-one mapping between the categorical variable T_{total} and the trip time should be performed. A risk profile for

TABLE I
SUMMARY OF DRIVING BEHAVIORS

Behavior index	Behavior	Description
1	Excessive speeding	Exceeding safe speed/speed limit
2	Inexperience or unfamiliarity	Apparent general inexperience driving, unfamiliarity with a vehicle or a roadway
3	Avoiding an object	Avoiding a vehicle, pedestrian, or an object
4	Sudden braking	Sudden or improper stopping on a roadway
5	Right-of-way error	Right-of-way error due to decision or recognition failures, or an unknown cause
6	Driving slow	Driving slowly in relation to other traffic or below speed limit
7	Improper reversing	Improper backing up due to inattentiveness or other causes
8	Illegal or unsafe lane change or turn	Any improper or illegal lane change or turn
9	Aggressive driving	Such as aggressive acceleration or aggressive lane changing
10	Signal or sign violation	Violation action at traffic signs or signals
11	Safe	No evidence/presence of risky behavior
12	Fatigue	Drowsiness, sleepiness, and fatigue
13	Negligence	Includes improper or failure to signal, and driving past dusk without lights

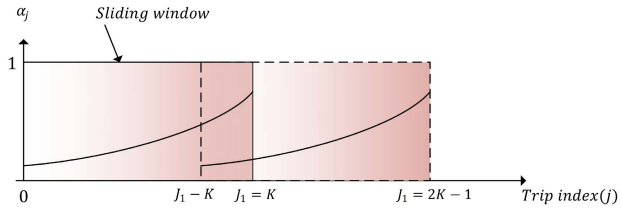


Fig. 4. Weighting function of the risk profile.

a certain driver (RP_{ID}) can then be expressed in terms of the weighted average score over the last K trips:

$$RP_{ID} = \sum_{j=T_1-K}^{j=J_1} \alpha_j \times Score_{ID}(j) \tag{7}$$

where

$$\sum_{j=T_1-K}^{j=J_1} \alpha_j = 1 \tag{8}$$

α_j is the weight associated with the j_{th} trip of the last K trips and can take a shape of an exponentially moving average function to give more weight for recent trips as being depicted in Figure 4.

IV. DATA FILTERING AND PRE-PROCESSING

A. Feature Engineering

As mentioned earlier, thirteen driving behaviors are identified and utilized to extract drivers' feature vectors F_{ID} to train and validate the proposed risk prediction models. Based on the adopted selection criteria, the selected behaviors are comprehensive and mutually exclusive in nature. They are chosen according to the following procedure:

1) In the SHRP2 dataset, driving behaviors are classified into 54 unique behaviors, spanning all possible driving behaviors. In the dataset, the three most identifiable behaviors inside the event time frame are recorded. For simplicity, only the most dominant behavior is chosen, which makes behaviors mutually exclusive for a given event ($P(B_i \cap B_j)_k = 0$, where $P(B_i \cap B_j)_k$ is the probability of the simultaneous occurrence of behaviors B_i and B_j at event k).
2) Behaviors that can be classed under the same category are combined to increase features' importance. Merging behaviors was an iterative process that included a compromise between reducing the models' over-fitting and avoiding the too broad generalization of behaviors resulting from merging too many behaviors in one "general" behavioral category. We initially attributed the over-fitting problem to there being a relatively small number of samples in some of the original behavior categories. Following our behavior merging process, which significantly enhanced the over-fitting performance, such behaviors - due to their rarity in the dataset - were proven to be a cause for over-fitting. At each *behavior merging* iteration, the classification/regression model is tested for over-fitting by comparing the model's train and test performances. As long as the model's performance is improving, additional behaviors with lower number of samples are merged with their corresponding "more general" behavioral categories. The "general" behavioral categories are chosen as to avoid overlap between them and to avoid the broader generalization that makes such behavioral classifications meaningless (e.g., good/bad behaviors). For instance, excessive speeding behavior is clearly distinct from sudden braking, slow driving, improper reversing, etc. An example of merged behaviors is the merging of: "*Driving slowly: below speed limit*" **and** "*Driving slowly in relation to other traffic: not below speed limit*" behaviors under the general behavioral category of "Driving slow". By following the same procedure for other behaviors, a total of 13 behavioral categories are identified.

The initial dataset is then formulated as shown in equation 9.

$$\left[\begin{array}{c|ccc|c} \text{Driver} & B_1(\%) & \dots & B_M(\%)\ T_{total} & \text{Outcome} \\ \hline 1 & & F_1 & & P(Risk|F_1) \\ \vdots & & \vdots & & \vdots \\ N & & F_N & & P(Risk|F_N) \end{array}\right] \tag{9}$$

The initially formulated features are further processed to enhance the performance of the models. Third order polynomial non-linear terms of the original features were added to increase models' flexibility. Moreover, to capture the interactions between the initially formulated features (i.e., the joint effect of features on risk), features' third order interaction terms were generated. Considering only three original features (f_1, f_2, f_3), their third order transformation is equivalent to (1, f_1, f_2, f_3, f_1^2, $f_1.f_2$, $f_1.f_3$, $\ldots$, $f_1.f_2.f_3$). With the large number of transformed features (i.e., 680), a feature extraction process was needed to reduce the feature space dimensionality. Such a process was crucial to enhance models' over-fitting performance and to minimize their training/testing processing time. For these reasons, Principal Component Analysis (PCA) technique was applied. As a result, the set of features was significantly reduced to a new set of features (often called principal components) that were still able to represent most of the variability in the data.

B. Data Filtering

Feature engineering process was followed by data filtering. The purpose of the filtering process was to remove cases (i.e., drivers) which did not have enough data to represent their behavioral patterns (i.e., insufficient number of baseline events). Such cases contributed in the models' irreducible error which is caused because of the limitation in the dataset. A sensitivity analysis was applied to find the minimum number of events (E_{best}) a driver should have without being filtered out from the dataset. Threshold values, which represent different numbers of captured events for each driver, in the interval [4, 10] were experimented and models' performances quantified in terms of Mean Square Error (MSE) were recorded for each threshold value. The trade-off was to find the best models' performance in terms of their MSE without losing too much data which can decrease a model's reliability. Having a marginal MSE enhancement in the proposed models' performance with a threshold value greater than 6, an $E_{best} = 6$ was adopted as a filtering criteria. Figure 5 depicts the histogram distribution for the number of captured events for all drivers contributed in the SHRP2 project.

After the filtering process, 29% of the cases were excluded. Despite the large number of filtered cases, there were still enough cases (i.e., 2007 cases) for the models' to be trained on and to be able to generalize with a high level of accuracy on test cases as shown in section VII. In real life applications, the rate at which behaviors are detected is supposed to be high enough to represent the behavioral patterns of all drivers. Detected behaviors during a certain trip will be augmented in the risk scoring function and drivers will be profiled according to the expected risk of their behavioral pattern.

Filtered data are highly skewed to the left as can be deduced from figure 5. In section V, different machine learning algorithms are investigated and compared to obtain the best modeling performance for such skewed data.

C. Feature Scaling

Classification bounds for machine learning algorithms such as SVM and KNN are obtained by calculating the Euclidean distance between feature vectors. These algorithms will not work efficiently without feature normalization [35]. This is because if one of the features has a broader range of values than the others, the aforementioned algorithms will be biased to this specific feature since the minimum distance will be governed by that feature. As a result, it is always a good practice to have the same range of values for all features. In this work, feature normalization was applied to the SVM, KNN, ELM, and ANN based models. The following normalization equation was adopted:

$$\hat{X} = \frac{X - \mu_x}{\sigma_x} \tag{10}$$

where X is the raw feature vector, $\hat{X}$ is the normalized feature vector, μ_x is the mean of X, and σ_x is the standard of deviation of X.

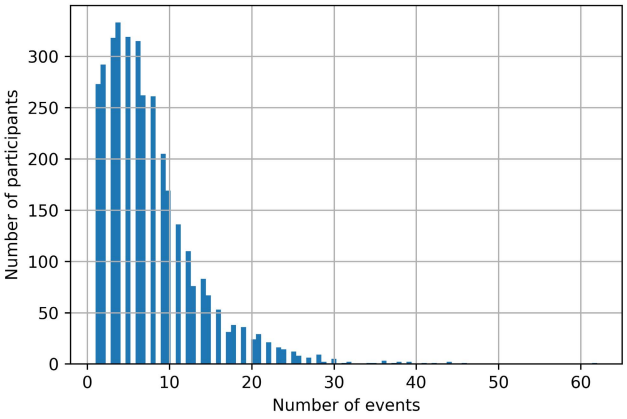


Fig. 5. Histogram distribution for the number of captured events for drivers in the SHRP2 dataset.

D. Dependent Variable (Output) Selection

In this work, the risk prediction problem is formulated using two different approaches. Initially it is formulated as a binary classification problem according to the following expression:

$$Outcome_{ID} = \begin{cases} 1, & \text{if } P(Risk|F_{ID}^{th}) > p_{th} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where p_{th} is a threshold risk probability above which the driver is considered risky. The value of p_{th} can vary according to the driving risk tolerance. In this work, a value $p_{th} = 0$ is adopted.

The problem is then formulated as a regression problem where $Outcome_{ID}$ takes the soft values of $P(Risk|F_{ID}^{th})$:

$$Outcome_{ID} = P(Risk|F_{ID}^{th}) \tag{12}$$

Each of these two risk prediction representations is important according to the domain in which driver profiling is applied. For instance, in the fleet management domain where drivers are warned if their behavior entails risky maneuvers, the binary classification would be more sensible. On the other hand, for insurance applications the adoption of the classification scenario may cause the loss of important information due

to the generality that classification entails, since risk scores are averaged over several trips.

V. Selection and Customization of Algorithms

In order to tackle the risk prediction problem, a performance comparative study is performed on several selected and customized machine learning algorithms. In this section, we present the six machine learning algorithms selected and the choice of their hyper-parameters. The selection of the candidate algorithms was motivated by two main factors:

1) The non-linearity of the feature space which motivated the sole use of non-linear classifiers/regressors.
2) The inter-dependencies between the risk prediction features. Inter-dependencies are clearly present between the initial behavioral features $\left(i.e., (B_i(\%)\right)$ because their values are complementary to each other. This occurs because they represent the rate at which each behavior occurs and they add up to one for each driver. So the increase/decrease in one feature will be reflected in the decrease/increase in other features. To show this mathematically, a vector that shows the correlation coefficients between the first and the rest of the features is displayed below:

$$Corr_{1,i} = [1.0, -0.565, -0.481, \cdots, 0.114] \quad (13)$$

Note that the absolute values of the correlation coefficients is larger than zero, which reflects the inter-dependencies between features. This led us to exclude algorithms that assume features' independency such as the Naive Bayes algorithm.

The selected algorithms and the choice of the adopted hyper-parameters are presented next.

A. K-Nearest Neighbors (KNN)

Despite its simplicity, the KNN algorithm has been successfully applied in several classification and regression based applications. The algorithm labels a new feature vector by applying a majority voting rule on the labels of its nearest neighboring samples, where neighbors are found by calculating their distances from the new feature vector. [36]. Distance is calculated using different measures such as: the Chebyshev distance (L_1-*Norm*), the Euclidean distance (L_2-*Norm*), and more generally the Minkowski distance (L_p-*Norm*). In the context of the proposed framework, for a feature vector $F_{ID} \in \mathbb{R}^{M+1}$, the Minkowski distance between two feature vectors is defined as:

$$D(F_l, F_m) = \left(\sum_{i=1}^{M+1} |F_l(i) - F_m(i)|^p \right)^{(\frac{1}{p})} \quad (14)$$

The choice of the optimal number of neighbors (K) as well as the Minkowski distance parameter p depends on the data distribution and the feature space size. This is usually considered a heuristic optimization problem and is beyond the scope of this paper. However, for the choice of K, we tested odd values to avoid tied votes [37]. Also, we noticed that performance does not improve for $K > 5$ and a performance degradation occurs for $K > 11$. Consequently, a K value of 5 has been adopted. Concerning the other hyper-parameter p, large values are usually chosen if the feature space is large. Since the feature space of our problem is relatively low (only 13 features), the commonly used L_2-*Norm* distance (i.e. $p = 2$) has been used.

TABLE II
SVM Adopted Hyper-Parameters

Parameter	Values				
k	Linear	Polynomial	Gaussian radial basis	-	-
C	1	5	10	50	100
γ	0.01	0.05	0.7	0.1	0.2
p	2	3	4	5	6

B. Support Vector Machines (SVM)

SVM is a very popular machine learning algorithm that has been applied in different classification and regression problems. For instance, it has been applied in bioinformatics, road anomalies and driver behavior classification, and other wide range of applications [21]. The algorithm is based on the margin-maximization principle detailed in [38]. In order to achieve the best classification performance, different hyper-parameters need to be optimized. Grid search technique is adopted in this work to find the best combination of hyper-parameters. Grid search technique performs an exhaustive search over a grid of hyper-parameter values to find the best set of hyper-parameters. It is often used when limited number of hyper-parameters with limited search spaces need to be optimized. The results presented in table II represent the effort of three rounds of grid-search tuning where the search space for each hyper-parameter is confined at each round. We used the area under the Receiver Operating Characteristic (ROC) curve as a performance metric for grid search.

In this work,the optimization is performed over four hyper-parameters which are the regularization parameter C, the kernel function k, the polynomial degree p, and the sensitivity parameter γ. The parameter C is necessary to avoid the overfitting problem. It is a regularization parameter where a small value indicates high regularization and a large value reflects low regularization. It determines which training samples are considered as outliers. The k parameter specifies the type of the kernel function. The kernel function transforms the data such that data separation would be easier. The usefulness of this function depends on the distribution of the data, which is not always intuitive for data with high dimensions (multi-variable data). So, we tried three popular kernels which are linear, polynomial and Gaussian radial basis functions. γ is a sensitivity parameter to measure the similarity between the feature vectors. For instance, if γ is large, feature vectors will be considered similar only if the Euclidean distance between them is small. A more detailed explanation of these hyper-parameters are found in [39]. Table II shows the investigated hyper-parameters and the best combination is shaded. Finally p is the order of the polynomial kernel.

TABLE III
DT AND RF ADOPTED HYPER-PARAMETERS

Decision Tree					
Parameter	*Values*				
MD	12	14	16	18	20
Random Forest					
Parameter	*Values*				
MD	20	22	24	26	28
n_estimators	80	90	100	110	120

C. Decision Trees (DTs) and Random Forests (RFs)

Unlike KNN and SVM, DT and RF classifiers (or regressors) do not rely on the minimum distance criterion. A decision tree finds a splitting point on the best predictor's histogram and incrementally builds a tree-structured classifier (or regressor) in a top-down fashion. Decision nodes in each level are chosen such that the entropy is minimized (or equivalently the information gain is maximized). For instance, the top most decision node (the best predictor node) will have the highest homogeneity as it will maximize the information gain. RFs are very similar to DTs except they use a multitude on decision trees on random subsets of the data to reduce overfitting, which is a common DT problem. DTs and RFs are very intuitive and because of their trackable structure, the importance of each feature can be easily measured, which can be very insightful in many applications, see [40] for detailed information.

To achieve the best risk prediction performance, we optimize the maximum depth (MD) of the decision tree and the number of trees in the random forest estimator (n_estimators). Performance is also calculated in terms of the area under the ROC curve. Table III depicts the values that are used for the two hyper-parameters and the best combination is shaded. The values presented herein are the fine-tuned values of the last round of trials.

D. Deep Neural Networks (DNNs)

Using multiple sequential computational layers, Deep Neural Networks (DNNs) learn data representation through multiple levels of abstraction as depicted in figure 6. Based on the original features from the input layer, each hidden layer creates more complex features based on interactions of features from a previous layer. DNNs do not need feature engineering since features with higher levels of abstraction are naturally extracted during back propagation. Using back-rpopagation algorithms such as Stochastic Gradiant Descent (SGD), ADAM algorithm, RMSProp, Limited memory BFGS, etc., DNNs learn how the internal parameters between every two layers represented by the matrix $W^{(i)}$ should change to minimize a chosen aggregated loss function $L(W)$, where $W = [W^{(1)}, W^{(2)}, \ldots, W^{(H)}]$ for a DNN with H hidden layers.

In this work, a customized **feed-forward** DNN was adopted. The adopted DNN's hyper-parameters include the rate at which the weights are updated at each iteration (learning rate α), Momentum which helps in preventing oscillations around the cost function global minimum, the number of hidden layers, the number of hidden units per layer, the regularization parameter ($L2$ penalty) which helps in preventing over-fitting, the number of epoches, the optimization algorithm for updating the network's weights and the choice of the activation function. Due to the large number of hyper-parameters, Grid search was discarded as it is considered a computationally inefficient hyper-parameters' optimization technique in such cases. So, we applied the random search technique to find the optimal set of hyper-parameters which are displayed in table IV.

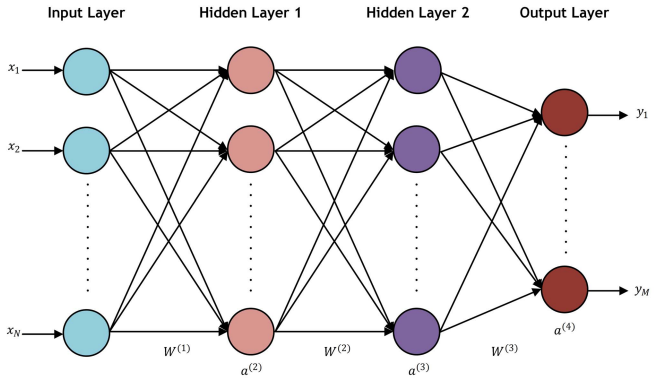


Fig. 6. An example of a DNN with two hidden layers.

TABLE IV
DNN ADOPTED HYPER-PARAMETERS

Parameter	**Adopted Value**
Learning rate (α)	0.001
Momentum	0.9
Number of hidden layers	5
Number of hidden units	5
$L2$ penalty	0.0001
Number of epochs	200
Optimization algorithm	Limitied-memory BFGS
Activation function	RELU

TABLE V
ELM ADOPTED HYPER-PARAMETERS

Parameter	**Adopted Value**
Number of hidden units	100
Activation function	Sine

E. Extreme Learning Machines (ELMs)

A special case of Artifical Neural Networks (ANNs) is the Extreme Learning Machines (ELMs). An ELM is a single layer feed-forward ANN with a random number of hidden neurons and Ordinary Linear Least Squares (OLS) algorithm applied to find the network weights' matrix through a single optimization step. ELMs take much less training time than ANNs trained through back-propagation and can give comparable results. The only two hyper-paramters in ELMs are the number of hidden units, and the activation function. Table V shows the chosen ELM's hyper-parameters which were found through grid search.

VI. Performance Assesment Metrics

To quantify the quality of the algorithms presented in the previous section, different performance assessment metrics for classification and regression models have been adopted.

A. Classification Models

1) Accuracy: is one of the most often used measures for assessing the performance of machine learning algorithms. It measures the overall performance of a classifier and is expressed as:

$$Accuracy = \frac{T_p + T_N}{T_p + T_N + F_p + F_N} \tag{15}$$

where T_p, T_N, F_p, F_N are respectively referring to the number of true positive, true negative, false positive and false negative samples.

2) F1-Score: also called the harmonic mean of precession and recall. It gives an insight on the combined performances of precession and recall. It is defined as:

$$F1 = 2 \times \frac{Precession \times Recall}{Precession + Recall} = \frac{2T_p}{2T_p + F_p + F_N} \tag{16}$$

3) ROC Curves: reflect the classification performance of a binary classifier as we change a threshold on the classifier soft probability values. It is a comparison of the recall (i.e. the true positive rate) and the false positive rate as the threshold is altered. Using ROC curves, the performance of a classifier is measured mainly in terms of the area under the curve (AUC), where the better the classifier performs, the closer the AUC gets to 1.

B. Regression Models

Let $\hat{Y}$ be a vector of N_T predictions containing the predicted risk probabilities for N_T drivers, and Y is the test vector that contains the true N_T risk probabilities.

1) Mean-Square Error (MSE): MSE is defined as the squared sum of the averaged differences between predictions and true labels. It can be expressed as:

$$MSE = \frac{1}{N_T} \sum_{1}^{N_T} (Y_i - \hat{Y}_i)^2 \tag{17}$$

2) Mean-Absolute Error (MAE): MAE is defined in terms of the absolute deviation between true and predicted values. This is mathematically written as:

$$MAE = \frac{1}{N_T} \sum_{1}^{N_T} |Y_i - \hat{Y}_i| \tag{18}$$

3) R^2 Value: also known as the coefficient of determination is another important statistical measure for assessing the performance of prediction models. It measures how much variance in the test vector Y the model can describe. It is computed using this formula:

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} \tag{19}$$

where $SS_{Regression}$ and SS_{Total} are, respectively, the squared sum of the regression error and the squared sum of the total error. They are mathematically expressed in equations 20 and 21:

$$SS_{Regression} = \sum_{1}^{N_T} (Y_i - \hat{Y}_i)^2 \tag{20}$$

$$SS_{Total} = \sum_{1}^{N_T} (Y_i - \bar{Y})^2 \tag{21}$$

where $\bar{Y}$ is the mean of the test vector Y. So, an R^2 value of zero indicates that the model does not perform better than a line that passes through the mean of the test vector Y. On the other hand, an R^2 value of 1 indicates that the model can explain all the variability of the test vector Y around $\bar{Y}$.

VII. Results and Discussion

This section presents the performance results of the algorithms described in section V. The algorithms were implemented in Spyder (Python 3.6) integrated development environment (IDE) using the Scikit-Learn Library for Machine Learning and Data Mining.

A. Training and Testing Splitting Methodologies

Two training and testing splitting methodologies have been adopted to train and validate the models.

1) *General Splitting Approach*: this is the common method used for choosing a randomly selected portion of the dataset for training and leaving the remaining dataset for testing. The splitting ratio usually depends on the amount of collected data and the application. In this work, 70% of the dataset is utilized for training. As a result, 1, 404 training samples and 603 testing or validation samples are used after fixing the random seed for the splitting process to a constant number.
2) *K-fold Cross Validation*: in this approach, the entire dataset is randomly divided into K equally sized partitions. In each training/ testing cycle, a single partition is kept for testing and all the remaining partitions are used for training. Training and validation is performed K times with each of the single partitions used once for testing. The mean and standard of deviation of the results can then be obtained to have more a statistical reflection on the model's performance. This approach is superior over the first approach since all data samples are utilized for both training and testing. In this work, a *10-fold cross-validation* is adopted for all models.

B. Classification Results

ROC curves for the six aforementioned algorithms using the general splitting approach are depicted in figure 7. As this figure illustrates, the RF algorithm produces the best AUC results among all other classifiers followed by the DNN. Specifically, RF produces the highest true positive rate for small false positive rates (i.e., $FP < 0.1$).

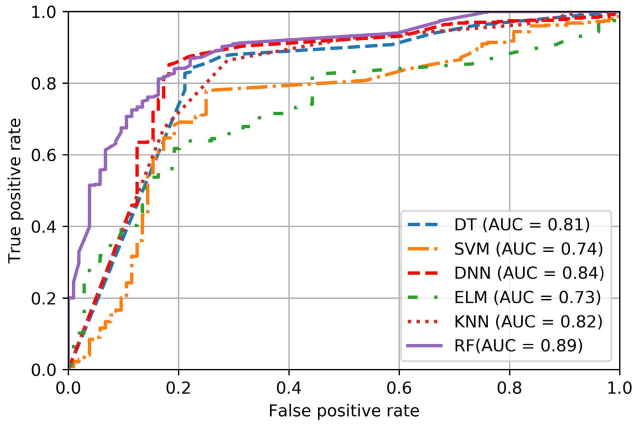


Fig. 7. ROC Curves for DT, SVM, DNN, ELM, KNN and RF classifiers using the general splitting approach.

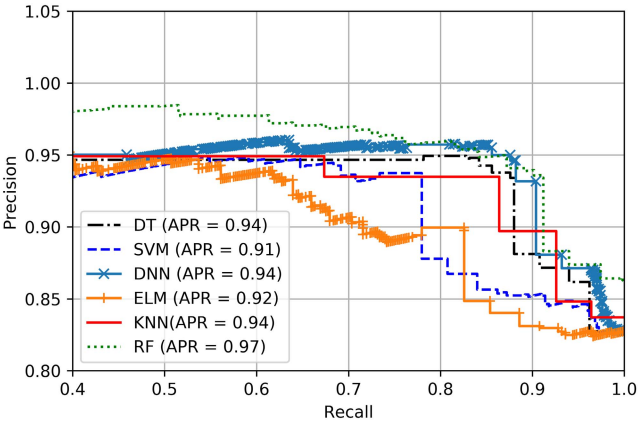


Fig. 8. Precision-Recall Curve for DT, SVM, DNN, ELM, KNN and RF classifiers using the general splitting approach.

Another measure of performance is the precision-recall curve. It gives useful insight on a classifier's performance for unbalanced labels. Figure 8 shows the precision-recall curves for the six algorithms. Again, the RF classifier clearly outperforms all other classifiers with an average precision of 97%.

A summary of the remaining performance assessment results using the general splitting approach is shown in Table VI. The table shows a consistency in performance superiority for RF classifier over the other five classifiers in all measures. RF achieves an accuracy of 87% and an F1-score of 0.93.

Figure 9 depicts the performance results using the 10-fold cross validation approach. The shown figure displays the variation in performance metrics distributions for the six classifiers using whisker plots. Points outside whisker plots' range $[Q1 - 1.5 * IQR, Q3 + 1.5 * IQR]$ are considered outliers, where $Q1$ and $Q3$ are respectively the first and third quartile values of the whisker plot, and IQR refers to its interquartile range (i.e., $IQR = Q1 - Q3$). Figure 9a shows that the RF classifier has an accuracy that ranges between 88.5% and 92.2% with an average accuracy 90%. The figure shows that the RF classifier outperformed the other five classifiers

TABLE VI

CLASSIFICATION PERFORMANCE RESULTS USING THE GENERAL SPLITTING APPROACH

	Performance measure		
Algorithm	Accuracy (%)	F1-score	ROC curve AUC (%)
DT	84.1	0.910	81.3
SVM	81.5	0.896	72.3
DNN	85.4	0.917	84.2
ELM	81.6	0.900	73.6
KNN	81.9	0.895	80.7
RF	86.9	0.926	89

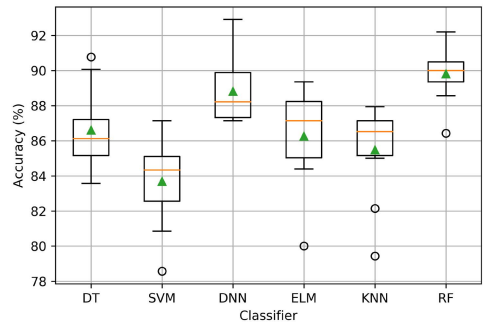
TABLE VII

CLASSIFICATION PERFORMANCE RESULTS USING 10-FOLD CROSS-VALIDATION

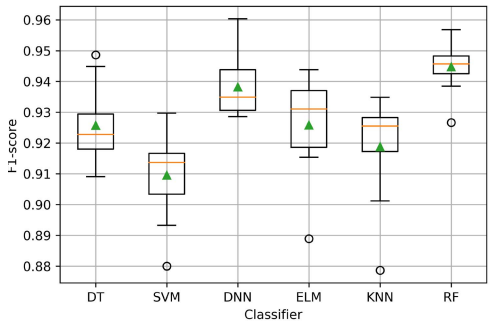
	Performance measures		
Algorithm	Average accuracy (%)	Average F1-score	Average ROC curve AUC (%)
DT	86.6	0.926	78.6
SVM	84	0.910	75
DNN	88.8	0.938	85.4
ELM	86.2	0.926	77
KNN	85.5	0.92	80
RF	90	0.945	87.5

in the average sense and in its performance consistency over different training/testing samples. Similar conclusions can be drawn from figures 9b and 9c where the superiority of the RF classifier is consistently evident. A summary of the results is shown in Table VII.

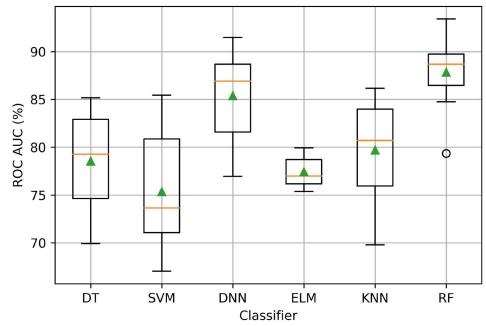
Two important observations can be made from the results. The first is the superior performance of the RF classifier over the DNN, whereas the second is the inferior performance of the SVM when compared to other classifiers. Concerning the first observation, despite the proven modelling power for DNNs, they seem to show their full potential when dealing with highly non-linear modelling problems with large number of features and a very large number of training samples (big data). A possible reason of why the RF outperformed the DNN in this classification problem may be attributed to the size of the utilized dataset (intermediate size) and the relatively small feature space since only the 14 original features were used to train the DNN. With regards to the second observation, the poor performance of SVM in comparison to other classifiers is attributed to two main reasons. The first is the imbalanced classes in our classification problem since we have more positive labels, and secondly that the performance of SVM is highly dependent on the optimization of its hyper-parameters, especially the kernel function. Although different SVMs were trained on various kernels during the hyper-parameters' optimization as mentioned in section V, they still performed poorly when compared to other algorithms. From our experience, bagging algorithms (e.g., Random Forests) almost always outperform SVM for intermediate data-sets with a relatively low number of features (e.g., the utilized data-set) unless a kernel that reflects the features' distribution is found, which is a computationally inefficient process (i.e.,



(a) Accuracy

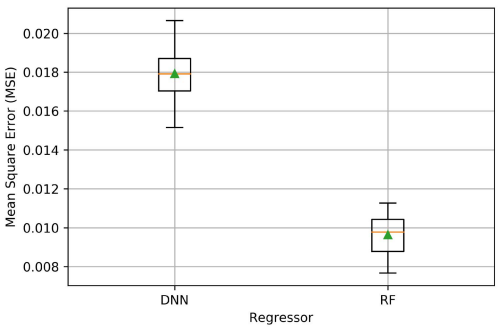


(b) F1-score

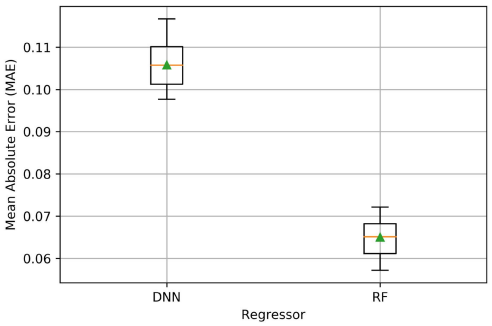


(c) ROC AUC

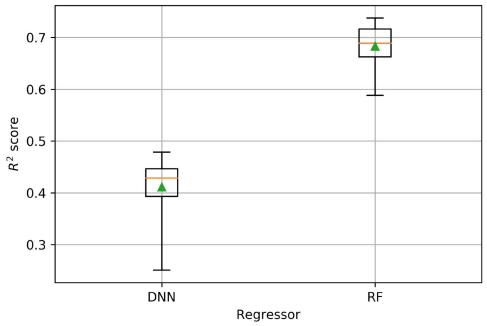
Fig. 9. Whisker plot for accuracy, F1-score and ROC AUC performances using 10-fold cross-validation.



(a) MSE



(b) MAE



(c) R^2 score

Fig. 10. Whisker plot for MSE, MAE and R^2 performances using 10-fold cross-validation.

the computational complexity for training an SVM is between $O(n^2)$ and $O(n^3)$ where n is the number of training samples).

C. Regression Results

1) Comparison Between Regressors: We present herein the comparison results between DNN and RF regressors as they are the best two performing algorithms in the classification context. Table VIII shows the MSE, MAE and R^2 performance results for DNN and RF regressors using the general splitting approach. Similar to classification results, a RF regressor seems to outperform DNN regressor in all performance measures. Most importantly, R^2 value for RF regressor is considerably higher with a difference gain of 25% over DNN regressor.

Figure 10 shows the MAE, MSE and R^2 performance results of DNN and RF regressors. Again RF regressor outperforms DNN regressor in terms of consistency over different testing samples and in terms of its average performance. Particularly, DNN regressor seems to have very inconsistent R^2 results with a relatively small mean when compared to RF regressor. A summary of the results is shown in Table IX.

TABLE VIII

PREDICTION PERFORMANCE RESULTS USING GENERAL SPLITTING APPROACH

	Performance measures		
Algorithm	MSE	MAE	R^2
DNN	0.015	0.09	0.46
RF	0.008	0.05	0.71

Figure 11 depicts the prediction vs. true $P(Risk|F_{ID})$ for a random sample of 100 driverS in the test set using RF regressor. The figure shows the ability of RF regressor to correctly predict drivers' risk probabilities in most cases.

TABLE IX

PREDICTION PERFORMANCE RESULTS USING 10-FOLD CROSS-VALIDATION

	Performance measures		
Algorithm	Average MSE	Average MAE	Average R^2
DNN	0.018	0.105	0.41
RF	0.009	0.065	0.69

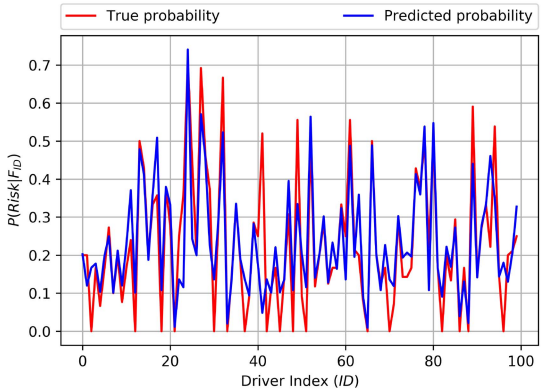


Fig. 11. Predicted vs. true risk probabilities for a sample of 100 driver using RF regressor.

TABLE X

COMPARISON BETWEEN PERFORMANCE RESULTS OF TWO RF MODELS USING CONVENTIONAL AND EXTENDED FOMS

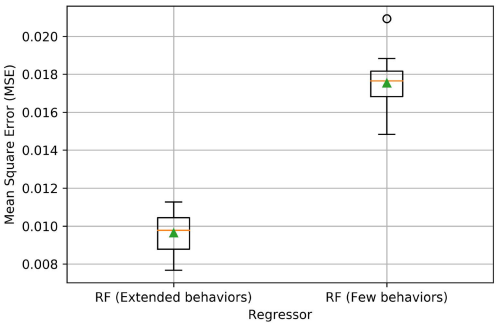
	Performance measures		
Algorithm	Average MSE	Average MAE	Average R^2
RF (Few FOMs)	0.018	0.097	0.43
RF (Extended FOMs)	0.009	0.065	0.69

2) Conventional vs. Proposed FOMs: We compare the performance of the RF model with the proposed predictors against its performance using the conventionally used FOMs that are usually adopted in the car insurance market, which are: excessive speeding, aggressive driving, sudden or improper braking and the total exposure time.

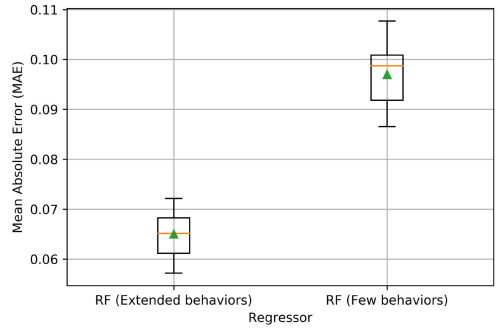
Figure 12 depicts the performance results of two RF models, one with the utilization of the proposed predictors (FOMs) and the other with the use of only the four conventional predictors that are used by car insurance companies. The results show a considerable difference between the two, where the model with the proposed FOMs is far superior. A summary of the comparison performance results is shown in Table X.

3) Test Cases: Two test cases are presented in Tables XI and XII. Table XI shows that the relatively low percentage of safe driving (i.e., B_{11}) for driver 1 resulted in high risk probability of 0.655 specially when combined with highly risky behaviors such as: illegal or unsafe lane change or turn (B_8), fatigue or negligence (B_{12}), excessive speeding (B_1), and aggressive driving (B_9). In this case, the proposed RF regressor was able to predict the risk probability with a very low MSE of 0.0021.

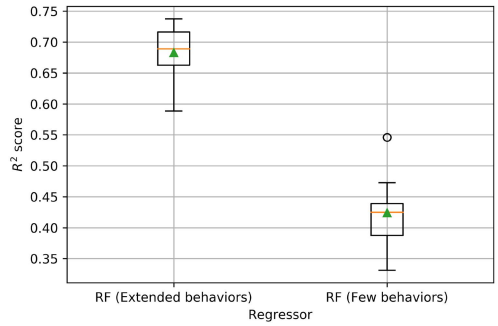
Table XII shows that the very high percentage of safe driving for driver 2 (i.e., 87.5 %) was the dominant factor



(a) MSE



(b) MAE



(c) R^2 score

Fig. 12. RF models' performances using conventional vs. proposed predictors.

TABLE XI

TEST CASE FOR DRIVER 1

F_1	B_1 (%)	25
	B_8 (%)	10
	B_9 (%)	20
	B_{11} (%)	35
	B_{12} (%)	10
	T_{total}	7
$P(Risk)$	0.655	
$P(Risk\|F_1)$	0.662	

in having a low risk probability of 0.125. Similar to the first case, the MSE value here is negligible.

An important finding in the presented and many other cases is that driving risk can be accurately predicted with only few events captured with an appropriate sampling

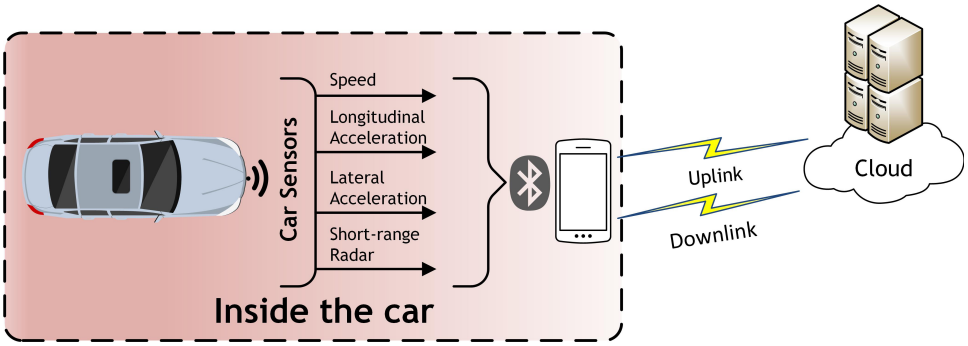


Fig. 13. Uplink: A driver's smartphone sends the collected OBDII, radar and its inertial measurements to the cloud for processing. Inside the cloud, behaviors are classified using sequence modeling and inputted to the proposed driver scoring model. Downlink: A trip score is issued to the driver on a per trip-basis.

TABLE XII
TEST CASE FOR DRIVER 2

F_2	B_8 (%)	6.2
	B_{11} (%)	87.5
	B_{12} (%)	6.2
	T_{total}	14
$P(Risk)$	0.125	
$P(Risk\|F_1)$	0.128	

time (i.e., balanced base-line events denoted here as T_{total}). That is because given the relatively low rate at which the baseline events were taken in the SHRP2 dataset [34], the risk prediction models' irreducible error was insignificant and a snapshot of the behavioral pattern of different drivers was enough to predict their long-term risk. Therefore, there is no need for a continuous driving data acquisition to determine the associated risk of a certain driver. This has its relevance in minimizing the consumed power of offloading driving data to the cloud server in a cloud-based profiling system and also in minimizing the computational cost for predicting driving risk.

Despite the insignificant models' irreducible error, more accurate results are anticipated given higher baseline events' sampling rate which should contribute to minimizing the models' errors.

One main limitation of the proposed RF-based driving risk prediction solution is that it is difficult to interpret. In other words, the use of the RF-based algorithm limits the ability to infer the associations between the original driving behaviors and risk when compared with simpler algorithms (e.g., linear regression and decision trees). For more reflections on the associations between driving behaviors and risk, there should be a compromise between a model's predictability and interpretability. Moreover, a risk prediction model may benefit from exploiting the spatial context of the driving behaviors using mapping techniques (e.g., GIS mapping) [41], [42]. Such mapping is expected to enhance the prediction performance and the interpretability of a model.

VIII. FUTURE WORK: CLOUD-BASED PROFILING SYSTEM

In real life profiling applications, the proposed risk profiling system can be hosted in a cloud as depicted in Figure 13. In the envisioned cloud-based profiling system, a smartphone application will serve as a hub in which real-time vehicle's network data (i.e., through OBDII units), the radar range data, and the smartphone inertial measurements are collected and forwarded to the cloud. Inside the cloud, such real-time data are leveraged to detect/classify driving behaviors through sequence modelling. Detected behaviors are then augmented in the proposed risk scoring function at the end of each driving trip. The calculated score is utilized to update the driver's risk profile and is sent back to the driver on a per trip basis.

A main challenge of the proposed cloud-based profiling system is the ability to maintain a reliable communication link between the vehicle of the subject driver and the cloud. Reliable communication should ensure the reception of the vehicular data, and hence the accurate classification of driving behaviors. In the case of communication disruption, a possible solution would be storing the non-received data in the subject driver's smartphone and re-sending data when communication is re-established. All stored data should be stamped by the trip ID number and resent to the cloud where a trip score for the subject driver is recalculated.

For enhancing the statistical significance of the proposed results, more data is required. Semi-supervised methods, which use labeled and unlabeled naturalistic driving data, can be used at a little time and cost compared to supervised techniques which necessitates labeling before using the data [43].

IX. CONCLUSION

In this paper, a novel data-driven risk scoring framework for driver behavior profiling applications is proposed. Six machine learning algorithms are selected, customized and compared to achieve the best risk prediction performance. Algorithms are applied on SHRP2 NDS which is the largest NDS dataset collected to date. Results show the high performance standards these algorithms can achieve in predicting risk probability with a performance advantage of the RF-based predictor. It was shown that the RF-based predictor can accurately model skewed data in which the histogram of the number of captured events per drivers is highly skewed to the left. This has practical significance in accurately predicting drivers' risk even for relatively short driving time. Good performance results are found consistent even with a relatively small number of

captured events. This finding is very useful in a cloud-based profiling system to lessen the amount of consumed power caused by data offloading, to reduce the computational cost for predicting driving risk and to minimize the time needed before warning risky drivers.

A comparison between two customized RF regression models, one trained with only few conventionally used predictors (FOMs) and the other trained with an extended set of proposed FOMs is established. The results show that the latter model outperforms the former in all performance measures as well as in performance stability over different sets of validation samples. Finally, given the successful results, the incorporation of the proposed system into a practical cloud-based driver profiling system is warranted. This system could be of great benefit to driver profiling companies in car insurance telematics and fleet administration domains.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers whose comments have greatly improved this article.

This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The findings and conclusions of this article are those of the authors and do not necessarily represent the views of the Virginia Tech Transportation Institute, SHRP 2, the Transportation Research Board, or the National Academy of Sciences.

REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[3] A. S. El-Wakeel, J. Li, A. Noureldin, H. S. Hassanein, and N. Zorba, "Towards a practical crowdsensing system for road surface conditions monitoring," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4672–4685, Dec. 2018.

[4] M. Hung, "Gartner insights on how to lead in a connected world," Gartner, Inc., Stamford, CT, USA, Tech. Rep., 2017.

[5] B. Hussain, Q. U. Hasan, N. Javaid, M. Guizani, A. Almogren, and A. Alamri, "An Innovative Heuristic Algorithm for IoT-Enabled Smart Homes for Developing Countries," *IEEE Access*, vol. 6, pp. 15550–15575, 2018.

[6] P. Kodeswaran, R. Kokku, M. Mallick, and S. Sen, "Demultiplexing activities of daily living in IoT enabled smarthomes," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[7] *Wearable Devices and the Internet of Things Mouser*. Accessed: Nov. 4, 2020. [Online]. Available: https://ca.mouser.com/applications/article-iot-wearable-devices/

[8] L. F. Herrera-Quintero, J. C. Vega-Alfonso, K. B. A. Banse, and E. Carrillo Zambrano, "Smart ITS sensor for the transportation planning based on IoT approaches using serverless and microservices architecture," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 17–27, Dec. 2018.

[9] (2015). *WHO Global Status Report on Road Safety*. [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/2015/e%n/

[10] T. S. J. Darwish and K. A. Bakar, "Fog based intelligent transportation big data analytics in The Internet of vehicles environment: Motivations, architecture, challenges, and critical issues," *IEEE Access*, vol. 6, pp. 15679–15701, 2018.

[11] G. Castignani, R. Frank, and T. Engel, "Driver behavior profiling using smartphones," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 552–557.

[12] P. Handel *et al.*, "Insurance telematics: Opportunities and challenges with the smartphone solution," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 57–70, Dec. 2014.

[13] V. Neale, T. Dingus, S. Klauer, J. Sudweeks, and M. Goodman, "An overview of the 100-car naturalistic study and findings," in *Proc. 19th Int. Tech. Conf. Enhanced Saf. Vehicles (ESV)*, Jun. 2005, pp. 1–10.

[14] K. L. Campbell, "The SHRP 2 naturalistic driving study," Transp. Res. Board, Washington, DC, USA, Tech. Rep. TR news 282, 2012.

[15] R. Eenink, Y. Barnard, M. Baumann, X. Augros, and F. Utesch, "UDRIVE: The European naturalistic driving study," in *Proc. Transp. Res. Arena (TRA)*, Paris, France, 2014.

[16] T. A. Dingus *et al.*, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 10, pp. 2636–2641, 2016. [Online]. Available: http://www.pnas.org/content/113/10/2636

[17] *SHRP2 NDS Data Access*. Accessed: Nov. 4, 2020.

[18] T. Chakravarty, A. Ghose, C. Bhaumik, and A. Chowdhury, "MobiDriveScore; A system for mobile sensor based driving analysis: A risk assessment model for improving one's driving," in *Proc. 7th Int. Conf. Sens. Technol. (ICST)*, Dec. 2013, pp. 338–344.

[19] "TD myadvantage safe driving discount insurance," TD Insurance, Ottawa, ON, Canada, Tech. Rep. [Online]. Available: https://www.tdinsurance.com/products-services/auto-car-insurance/my-advantage

[20] A. Abdelrahman, N. Abu-Ali, and H. S. Hassanein, "Driver behavior classification in crash and near-crash events using 100-CAR naturalistic data set," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[21] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Driver behavior classification at intersections and validation on large naturalistic data set," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 724–736, Jun. 2012.

[22] S. Choi, J. H. Kim, D. G. Kwak, P. Angkititrakul, and J. H. L. Hansen, "Analysis and classification of driver behavior using in-vehicle canbus information," in *Proc. Biennial Workshop DSP In-Vehicle Mobile Syst.*, 2007.

[23] S. Daptardar, V. Lakshminarayanan, S. Reddy, S. Nair, S. Sahoo, and P. Sinha, "Hidden Markov model based driving event detection and driver profiling from mobile inertial sensor data," in *Proc. IEEE Sensors*, Nov. 2015, pp. 1–4.

[24] C. G. Quintero M., J. O. Lopez, and A. C. Cuervo Pinilla, "Driver behavior classification model based on an intelligent driving diagnosis system," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 894–899.

[25] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 91–102, Dec. 2015.

[26] N. Arbabzadeh and M. Jafari, "A data-driven approach for driving safety risk prediction using driver behavior and roadway information data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 446–460, Feb. 2018.

[27] J. F. Jãnior *et al.*, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS ONE*, vol. 12, no. 4, Apr. 2017, Art. no. e0174959. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.01749%59

[28] L. M. Bergasa, D. Almeria, J. Almazan, J. J. Yebes, and R. Arroyo, "DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 240–245.

[29] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 387–392.

[30] C.-W. You *et al.*, *CarSafe: A Driver Safety App That Detects Dangerous Driving Behavior Using Dual-Cameras on Smartphones*. New York, NY, USA: ACM Press, 2012, p. 671. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2370216.2370360

[31] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 234–239.

[32] K. Tang, S. Zhu, Y. Xu, and F. Wang, "Modeling Drivers' dynamic decision-making behavior during the phase transition period: An analytical approach based on hidden Markov model theory," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 206–214, Jan. 2016.

[33] T. G. Brown *et al.*, "The effect of age on the personality and cognitive characteristics of three distinct risky driving offender groups," *Personality Individual Differences*, vol. 113, pp. 48–56, Jul. 2017.

[34] J. M. Hankey, M. A. Perez, and J. A. McClafferty. (Apr. 2016). *Description of the SHRP 2 Naturalistic Database and the Crash, Near-Crash, and Baseline Data Sets*. [Online]. Available: https://vtechworks.lib.vt.edu/handle/10919/70850
[35] A. B. Graf and S. Borer, "Normalization in support vector machines," in *Pattern Recognition*, vol. 2191, G. Goos, J. Hartmanis, J. van Leeuwen, B. Radig, and S. Florczyk, Eds. Berlin, Heidelberg: Springer, 2001, pp. 277–282.
[36] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Stat.*, vol. 46, no. 3, pp. 175–185, Aug. 1992.
[37] P. Hall, B. U. Park, and R. J. Samworth, "Choice of neighbor order in nearest-neighbor classification," 2008, *arXiv:0810.5276*. [Online]. Available: http://arxiv.org/abs/0810.5276
[38] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, no. 1, pp. 45–66, Mar. 2002.
[39] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
[40] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986. [Online]. Available: http://link.springer.com/10.1007/BF00116251
[41] J. Balsa-Barreiro, P. M. Valero-Mora, I. Pareja-Montoro, and M. Sánchez-García, "Proposal of geographic information systems methodology for quality control procedures of data obtained in naturalistic driving studies," *IET Intell. Transp. Syst.*, vol. 9, no. 7, pp. 673–682, Sep. 2015.
[42] J. Balsa-Barreiro, P. M. Valero-Mora, J. L. Berné-Valero, and F.-A. Varela-García, "GIS mapping of driving behavior based on naturalistic driving data," *ISPRS Int. J. Geo-Information*, vol. 8, no. 5, p. 226, May 2019.
[43] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1108–1120, Apr. 2016.

Abdalla Ebrahim Abdelrahman (Student Member, IEEE) received the B.Sc. degree in control and instrumentation systems engineering and the M.Sc. degree in electrical engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2010 and 2013, respectively, and the Ph.D. degree in computer and electrical engineering from Queen's University, Kingston, ON, Canada, in 2019. His research interests include driver behavioral modeling and profiling, intelligent vehicular systems, machine learning, and deep learning. His work has been published in IEEE flagship conferences and top-tier journals. He also serves as a TPC member and a reviewer for IEEE flagship conferences and journals.

Hossam S. Hassanein (Fellow, IEEE) is currently a leading authority in the areas of broadband, wireless and mobile networks architecture, protocols, and control and performance evaluation. His record spans more than 500 publications in journals, conferences and book chapters, and in addition to numerous keynotes and plenary talks in flagship venues. He received several recognition and best papers awards at top international conferences. He is also the Founder and the Director of the Telecommunications Research Laboratory (TRL), School of Computing, Queen's University, with extensive international academic and industrial collaborations. He is also a Former Chair of the IEEE Communication Society Technical Committee on Ad Hoc and Sensor Networks (TC AHSN). He was an IEEE Communications Society Distinguished Speaker (Distinguished Lecturer) from 2008 to 2010.

Najah Abu-Ali (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Jordan and the Ph.D. degree from the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada; specializing in resource management in computer networks. She is currently an Associate Professor with the Faculty of Information Technology, United Arab Emirates University (UAEU). Her research interests include modeling wireless communications, resource management in wired and wireless networks, and reducing the energy requirements in wireless sensor networks. More recently, she has strengthened her focus on the Internet of Things, particularly at the nano-scale communications level, in addition to vehicle-to-vehicle networking. Her work has been consistently published in key publications venues for journals and conference. She has further coauthored a Wiley book on *Next-Generation Wireless Technologies: 4G and Beyond (Computer Communications and Networks)*. She has also delivered various seminar and tutorials at both esteemed institutions and flagship gatherings. She has also been awarded several research fund grants, particularly from the Emirates Foundation, ADEC, NRF/UAEU funds, and the Qatar National Research Foundation.