ATVSA: Vehicle Driver Profiling for Situational Awareness

Rashid Khan, Neetesh Saxena, Omer Rana
School of Computer Science and Informatics
Cardiff University
Cardiff, United Kingdom
{*khanrm1, saxenan4, ranaof*}*@cardiff.ac.uk*

Prosanta Gope
Department of Computer Science
University of Sheffield
Sheffield, United Kingdom
p.gope@sheffield.ac.uk

***Abstract*—Increasing connectivity and automation in vehicles leads to a greater potential attack surface. Such vulnerabilities within vehicles can also be used for auto-theft, increasing the potential for attackers to disable anti-theft mechanisms implemented by vehicle manufacturers. We utilize patterns derived from Controller Area Network (CAN) bus traffic to verify driver "behavior", as a basis to prevent vehicle theft. Our proposed model uses semi-supervised learning that continuously profiles a driver, using features extracted from CAN bus traffic. We have selected 15 key features and obtained an accuracy of 99% using a dataset comprising a total of 51 features across 10 different drivers. We use a number of data analysis algorithms, such as J48, Random Forest, JRip and clustering, using 94K records. Our results show that J48 is the best performing algorithm in terms of training and testing (1.95 seconds and 0.44 seconds recorded, respectively). We also analyze the effect of using a sliding window on algorithm performance, altering the size of the window to identify the impact on prediction accuracy.**

***Index Terms*—Anti-theft, driver profiling, situational awareness, security, vehicle.**

1. Introduction

Profiling a vehicular system can improve situational awareness, predictive maintenance of a vehicle and as the basis to prevent theft. Increasing connectivity and automation in vehicles has led to vehicle-theft being a significant concern [1],[2],[3]. One of the major reasons is that these vehicles are exposed to a range of cyber risks that could be exploited by the attackers [4],[5],[6]. Although an Internet connection enables the availability of real-time traffic data, intelligent fleet management, car-sharing and autonomous driving – it also leads to new theft possibilities [7].

To reduce the number of auto-theft cases, many anti-theft technologies are being implemented across the world, but the cases of stolen vehicles are still increasing [8]. According to UK police statistics on auto crime, there were over 114K auto theft cases in England and Wales in 2018/19, an increase of 8K cases compared to the previous year [9]. Exploited vulnerabilities include increasing acceleration remotely, disabling the brakes of a vehicle, access to air conditioning and door locks and data injection through the telematics system [10]. The security of these vehicles will become more critical with the increased production of these vehicles. For instance, in 2014, thieves stole 6K+ vehicles using keyless techniques, which make up to half of all vans and vehicles stolen – with top of the range vehicles such as BMW and Range Rover making up 70% of all vehicles stolen in this way [11]. According to ITS Digest reports that there will be over 470 million connected vehicles by 2025 [12].

In this direction, the Controller Area Network (CAN) bus is researched with a range of machine learning algorithms for profiling the drivers and solve problems such as driver classification/identification, driver performance assessment, and individual driving style learning. In this work we use machine learning algorithms to analyze the driving patterns of each individual driver, to generate an alert if an unknown person is found to be the driver. The idea behind this approach is to improve the already existing models by experimenting with machine learning and obtain more precise the user driving patterns.

Our contributions are as follows:

1) We have analyzed in-vehicle Controller Area Network (CAN) traffic to authenticate (verify) a driver.
2) We propose "ATVSA", an approach to identify patterns for profiling the drivers. Our approach excludes identical and co-related features that reduce the overall processing time and improve detection performance.
3) We have validated our model using real driving data and demonstrated that semi-supervised machine learning is effective in detecting anti-theft. We used key features for the classification of drivers based on their behavioral characteristics.

2. Related Work

Data mining techniques use supervised learning utilizes labeled data for training purposes [13],[14],[15],[16],[17]. Table 1 shows work related to driver identification and profiling using machine learning algorithms, such as Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB) [18], K-Nearest Neighbor (KNN) [19], Multilayer Perceptrons (MLP), Fuzzy Neural Networks (FNN) [20] and K-means clustering [21]. Zhang et al. [22] use HMM to analyze unique driving patterns using an artificial simulator. They extracted different features related to steering and the accelerator, and classified different drivers with 85% accuracy. Meng et al. [23]

TABLE 1: Driver Classification and Profiling

Work	Data Set	# of Feature	Classification Algorithm	Accuracy
[23]	Driving simulation	3	HMM	99%
[27]	Sensor data	2	GMM	76.8%
[8]	CAN network data	4	GMM,HMM	25%
[29]	CAN network data	8	SVM, RF, NB, KNN	87%
[30]	Vehicle sensor & video stream	2	MLP, statistical, FNN	99%
[31]	CAN network data	4	Statistical	77%
[32]	Driving simulation	2	HMM	85%
[34]	Clustering	4	K-means	-

studied driving patterns using a game-based simulation, using HMM for classifying drivers based on features such as acceleration and wheel data. A simulated driving environment however does not model variable road conditions, weather, etc., and is therefore of limited benefit a real-world context.

Other studies on anti-theft detection use facial recognition [24], vehicle security systems using IoT devices [25], and authenticated access control for vehicle systems using driving license and fingerprinting [26]. However, such solutions do not utilize artificial intelligence to train the system model. Nishiwaki et al. [27] collected driving data with sensors installed on a Toyota Regius, which a number of drivers drove to support data collection. This real-world data capture is much more representative of actual usage. They applied the supervised learning algorithm Gaussian Mixture Model (GMM), and this model was able to differentiate between 276 drivers with 76.8% accuracy. Other work that analyzed data extracted from a CAN bus can be found in [28]. A single cable is required to extract CAN data making this a comparatively more economical method. CAN data from 9 drivers was collected by Choi et al. [8], and an HMM was subsequently used to classify drivers based on their unique driving patterns – although with a very low accuracy of 25%. Enev et al. [29] used multiple machine learning algorithms to enhance the driver identification model, using data collected from 15 drivers – with an accuracy of 87%. Wahab et al. [30] performed modeling of individual driving characteristics. They extracted features using GMM and wavelet transformation and showed that accelerator and brake pedal use are very efficient for profiling drivers. Kedar-Dongarkar et al. [31] classified drivers based on energy consumption by a vehicle. The authors categorized drivers into three types: aggressive, moderate and conservative, by analyzing driving patterns. Zhang et al. [22] (in addition to HMM), also used deep neural networks to extract unique driving patterns. They used multiple Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for driver identification. These studies showed that CAN data could be used for extracting driving patterns with data mining techniques [35]. All these models are trained using supervised learning. In this research, we propose a model based on supervised and semi-supervised training.

A number of shortcomings of supervised algorithms can be overcome by applying unsupervised learning, as they do not require the use of labeled data. Constantinescu et al. [33] clustered different driving styles using the Hierarchical Cluster Algorithm (HCA). Higgs and Abbas [34] used the K-means clustering algorithm for the identification of drivers. The suggested models show that driver identification could be performed using unsupervised algorithms. These unsupervised algorithms can create clusters for distinct driving patterns, but these algorithms cannot identify which cluster is related to a particular driver (or vehicle owner). If a cluster that can be associated with a vehicle owner cannot be identified, the model cannot be used for theft detection.

In previous studies, most of the data used were extracted from accelerators and brakes. Moreover, previous studies deployed complex pre-processing on the extracted features to enhance the performance of these features. To satisfy auto-theft detection requirements, we propose a hybrid model ATVSA to detect auto-theft using a semi-supervised approach.

3. Vehicle Environment and Associated Risks

This section presents the overall system model and attacks vector for vehicle theft in modern scenarios.

3.1. System Model

We present a system model for deploying security services to prevent vehicle theft, as shown in Fig 1. Our system consists of four major entities: vehicle owner, server, vehicle monitoring company and the vehicle itself. Data extracted from an internet-connected vehicle on driving behavior, using the On-Board Diagnostics (OBD)-II protocol, can be analyzed on a server located outside the vehicle, to generate an alarm to a vehicle monitoring company if an anomaly is detected.

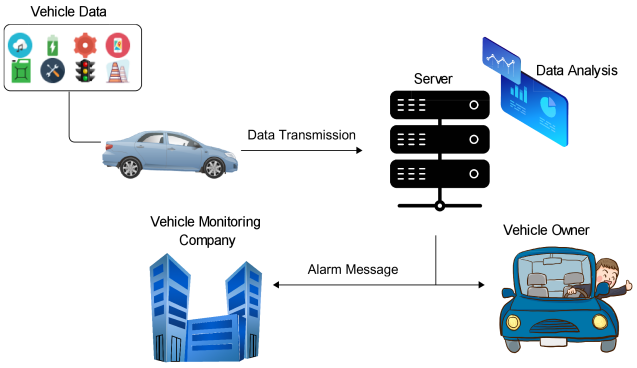


Figure 1: Vehicle security model in auto-theft.

3.2. Attack Vectors

We present a threat modeling with technical and operational adversarial capabilities against vehicle theft. Using

these methods, attackers can compromise the vehicle's system and manipulate or inject malicious data.

Technical Capability: we identify what thieves are likely to know about the vehicle and their ability to analyze the vehicle and develop malicious input for different I/O channels. Moreover, we assume that thieves have access to vehicle hardware in order to transmit messages (encoding suitable for any channel). We further assume that thieves are not capable enough to brute force complex shared secrets (e.g., symmetric encryption keys). In general, we assume that thieves can access information obtained directly by examining vehicle systems, similar to those targeted by the thieves in the past.

Operational Capability: includes the requirements for thieves to deliver any malicious input to a vehicles' input channel. Modern vehicles provide several physical interfaces that thieves can access directly or indirectly for accessing the vehicles' internal network. This includes On-Board Diagnostic-II (OBD-II), an infotainment system, and short and long-range wireless access. Our threat model assumes that thieves cannot have direct access, but using the OBD-II port, they can access the internal system and compromise the vehicles' internal network. Our research focuses on how driving behavior can be used to profile the drivers and how we can use the vehicle's different mechanical features for effective and accurate predictions.

4. Proposed Approach

In this section, we propose an anti-theft model, i.e. ATVSA that characterizes the driving patterns of drivers and identifies drivers on unique driving patterns using a semi-supervised learning approach. Figure 2 shows the proposed driver verification process based on the analysis of driving patterns. The proposed model of driver identification consists of five stages: data collection, data cleansing, feature selection, driver identification, and driver verification. When the driver starts driving the vehicle on the road, sensors within the vehicle start recording the data. Once the data is collected from the vehicles' sensors, the data is cleansed from any corrupt or inaccurate records. In the next stage, the cleansed data is converted into a new format, which has to be analyzed and different features selected are used to differentiate the drivers on unique driving characteristics. After features selection, the stage is set for applying different machine learning algorithms. For supervised learning, we apply four algorithms namely J48, Random Forest (RF), JRip, and PART, and the results obtained are compared with the pool of owner driver's data, along with utilizing unsupervised learning by employing the K-means/Canopy clustering algorithm to cluster the owner-driver data to create a pool of trusted driving styles.

The last stage is the verification process when validation data is provided to the proposed model, the model classifies the drivers on their unique driving patterns. When the thief data is introduced, the proposed model in its first instance of supervised learning deploys the classifiers for analyzing the accuracy. Note that the accuracy will be below the threshold value in the presence of the thief data. For the second instance, the system compares the data with the pool of owner data (i.e., clusters of owner data using unsupervised learning). The driving style of the thief driver will be not consistent with the pool of trusted driving data, a difference would be visible between the validation data of the thief and the selected owner-driver data. There will be a considerable gap between the data of both approaches, and with this gap, the proposed model generates an alarm to the owner-driver and/or vehicle monitoring company. Algorithm 1 provides an overview of our approach.

Algorithm 1: Proposed Approach for Anti-theft Detection

Output: Classification of drivers, whether the vehicle is driven by the owner or thief
Input: Dataset of 51 features extracted from the vehicle using CAN data

1 Input dataset compromising of 51 features into the WEKA tool for selecting the features that can be used for the analysis and classification;
2 After identifying the features that are extraneous and identical, 15 features have been chosen;
3 GainRatioAttributeEva method is applied to selected 15 features to derive the rank of the selected features;
4 Apply supervised learning classification algorithms and analyze the accuracy of the algorithms;
5 Apply an unsupervised learning clustering algorithm (i.e., K-means clustering and Canopy) clustering the owner-driver data to create a pool of trusted driving styles;
6 Compare the results of supervised learning with the results of unsupervised learning;
7 If there will be a significant gap between supervised learning and unsupervised learning, then report the case as a Theft Case.

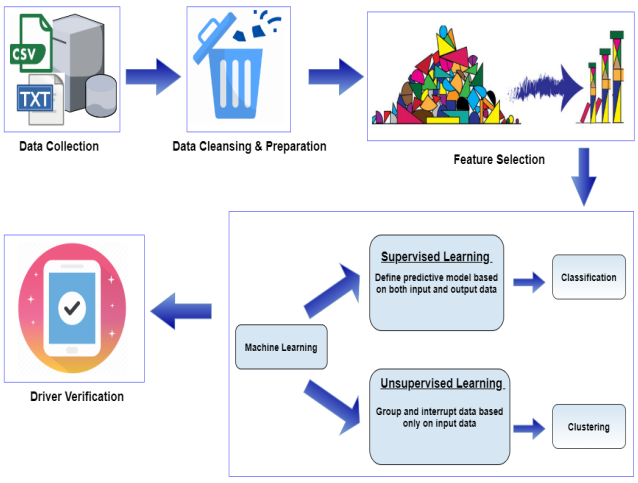


Figure 2: Proposed model - driver verification process.

4.1. Data Collection

Driving data has been extracted from the online KIA MOTORS Corporation (Seoul, South Korea) dataset [36]. Ten drivers participated in the experiment, and they drove

the vehicle on four paths in Seoul. The driving path consists of three ways: motorway, city way, and parking and with a total distance of 23 KM. All experiments were performed under the same time zone from 8 p.m. to 11 p.m. on weekdays. Each driver participating in the experiment completed two round trips for classification. The driving data was collected from different conditions, the city way has a speed breaker, traffic lights, etc. and the motorway has none. In parking space, it is required to drive slowly and cautiously. The total number of features that have been extracted is 51. The data that we used has a total of 94,401 records (recorded every second with a total size of 16.7 MB).

4.2. Designing Anti-Theft Approach

This subsection presents our anti-theft approach using machine learning algorithms and classifications. We highlight how the approach toward identifying anti-theft is designed. More specifically, the approach is designed to identify the ever-growing cases of anti-theft.

- *Supervised Learning*: We have the dataset that is labeled and have 51 features, each feature is relative to a class. The class is represented by the 10 drivers taking part in the experiment. All features belonging to drivers with associated labels represent the machine learning algorithm's input responsible for building the model from the analyzed data.
- *Unsupervised Learning*: We utilize the clustering algorithm for driver identification to satisfy the proposed requirements for semi-supervised learning. Using the K-means/Canopy clustering, we train only the owner data in this learning approach.

Output step: The model's output is a classification scheme belonging to either the car owner or the thief. Using the proposed approach, once the validation data of the thief is provided, the model in the first instance analyzes the accuracy of the classifiers (supervised learning). In the second instance, the thief driver data is compared with the pool of trusted driving data (unsupervised learning).

4.3. Feature Selection

In this section, we present our approach to process the features into new information that can be used for the identification and classification of drivers. To better understand the features, we have categorized the extracted features into three main *categories*: (1) *Transmission*: all features related to transmission and wheel, (ii) *Fuel*: all features related to fuel efficiency and pressure, and (iii) *Engine*: all features related to torque, engine and coolant temperature. We extracted 51 features from the vehicle's dataset.

A large amount of data was extracted from the CAN, so the selection of features was essential to train algorithms for achieving high accuracy against theft identification. In this first instance, we removed the features that had some correlation between them. Afterwards, we set up criteria for the removal of features from the dataset (by applying a set of rules).

- *Rule 1:* If any of the features contains a null value throughout the experimental driving.
- *Rule 2:* If a particular feature value collected from other drivers is indifferent. It implies no distinct values of a particular feature among the drivers participating in the experiment.
- *Rule 3:* If the feature's aggregated value and the standard deviation are zero for each driver. A zero value in our context of driver identification is meaningless as we analyze features that have distinguishing values among drivers participating in the experiment.

Rule 1 implies that there are errors while extracting data from the CAN, and a feature having missing or null values generates an error in the modeling. Rule 2 implies that there are no distinguishing characteristics of the feature among the drivers taking part in the experiment. Rule 3 implies that there is some data extraction error. If there is some kind of an unknown error, it is essential to check if the OBD-II is consistently recording zero values. Features satisfying the rules are removed to reduce the noise from the extracted data.

In our work, the driving data of different drivers are involved, and we filtered valuable features from the dataset of a total of 51 features. The hidden patterns can be used for profiling the drivers. Different *data preprocessing techniques* are involved as cleansing of data, integration of data, transformation, and reduction of data. These techniques can be used to improve the overall quality of the data. Preprocessing of data is essential for knowledge discovery as critical decisions are based on the quality of data.

Algorithm 2: GainRatioAttriEval Method

Output: A set {SA, RA, Wa}
where SA: selected attributes, RA: ranking of attributes, and Wa: weight of each attribute
Input: A set {CA, DA}
where CA: condition attribute of the driver's dataset and DA: decision attribute of the driver's dataset

1 Let ranking of attribute = Finite Ø;
2 Every attribute in the dataset, $a \epsilon$ CA – SA, the importance of condition attributes a and the Gain Ratio of (a, SA, DA) are calculated;
3 Choose the attributes from the dataset that maximizes the Gain Ratio (a, SA, DA), record the attributes as a, and SA$\leftarrow$ SA $\cup$ $\{a\}$;
4 If the Gain Ratio(a, SA, DA) >0, then SA$\leftarrow$ SA $\cup\{a\}$, go to step 2; else go to Step 5;
5 Selected attributes SA are chosen through the ranking value of attributes RA, which are based on the Gain Ratio;
6 Assign a weight to each attribute Wa, for the selected features SA.

The aim of applying *data reduction techniques* is to determine the data attributes that have the probability distribution of data classes as close as possible to the original probability distribution obtained using a dataset with all attributes. We use the GainRatioAttributeEval method for choosing the significant features within the dataset. This method evaluates the worth of an attribute by measuring the gain ratio with respect to a class [37]. The following formula calculates the Gain Ratio:

TABLE 2: Capabilities of different Attack surfaces

Feature	Category of Feature	Rank of Feature
Intake air pressure	Fuel	4
Fuel consumption	Fuel	11
Maximum indicated engine torque	Engine	5
Engine torque	Engine	6
Friction torque	Engine	3
Calculated load value	Engine	7
Engine coolant temperature	Engine	10
Transmission oil temperature	Transmission	2
Wheel velocity, front right hand	Transmission	14
Wheel velocity, front left hand	Transmission	12
Wheel velocity, rear left hand	Transmission	13
Torque converter speed	Transmission	15
Accelerator pedal value	Fuel	9
Activation of air compressor	Engine	8
Long term fuel trim bank1	Fuel	1

$GainR(Class, Attribute) = (H(Class) - H(Class|Attribute))/H(Attribute)$

Where H represents the Entropy. Entropy represents the randomness in the information being processed within the dataset. We have selected 15 features from 51 features extracted from the CAN dataset. Table 2 shows the extracted features and a specific rank of each of such features. This rank is computed by using the GainRatioAttributeEval method, as mentioned in Algorithm 2.

4.4. Feature Distribution

Figure 3 shows the box and whisker plots of features related to Fuel, Transmission and Engine, In-Take Air Pressure, Long Trim Fuel Bank 1, Friction of Torque, Maximum engine Torque and Transmission Oil Temperature. Considering several features, we have not included the box and whisker plots of all selected features, but similar consideration can be done for all 15 selected features. Through the box plot, we analyzed different values within the dataset that include minimum value, lower quartile (25%), median (50%), upper quartile (75%), and maximum value.

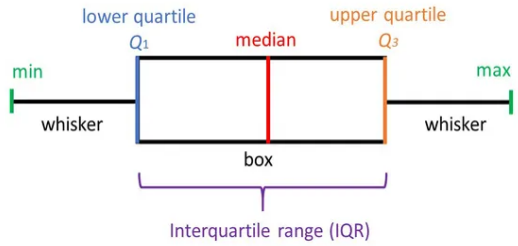
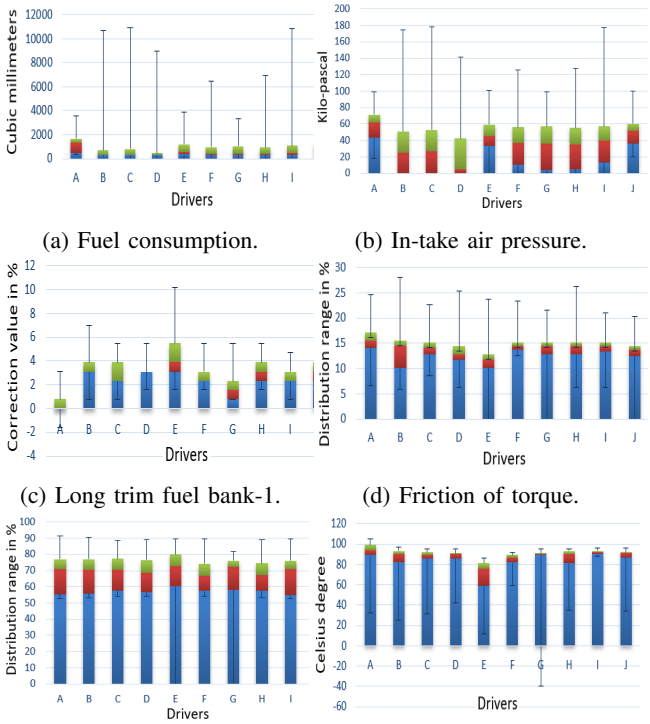


Figure 3: Layout of box and whisker plot.

Here, these box and whisker plots of different features highlight the driving characteristics of different drivers involved in the experiment. This helps us in setting up a platform for profiling the drivers. For calculation of values in box and whisker Plot, the following formulas will be used:



(a) Fuel consumption. (b) In-take air pressure.

(c) Long trim fuel bank-1. (d) Friction of torque.

(e) Maximum indicated engine torque. (f) Transmission oil temperature.

Figure 4: comparisons of different features; colour notation (Blue - first quartile, Red - second quartile, Green - third quartile); the line between Red and Green boxes - a median of the values.

$Box1 = FirstQuartile$ (1); $Box2 = Median - Quartile1$; $Box3 = Quartile3 - Median$ $Whisker\ Top = MaximumValue - Quartile3$ $Whisker\ Bottom = Quartile1 - MinimumValue$

From Figure 4a, it can be seen that the box plot of feature Fuel consumption has a range of 0-10,000 and it is measured in cubic millimeters (MCC). All drivers show similar kinds of box plots except driver A whose box plot is relatively more prominent as compared to other drivers. And the feature Fuel consumption can be correlated with another collected feature, Acceleration Pedal value (not in the selected list of 15 features) as more pressure on the acceleration pedal will increase the vehicles' speed, but on the other hand, the vehicle will consume more fuel. The feature of In-take air pressure (Figure 4b) has a range of 0-255 that is measured in Kilo-pascal (kPA). From the analysis of the box plot of 10 divers, it can be seen that the engine for drivers A, E, and J inhale similar air pressure (45 to 60 kPA). Furthermore, drivers B, C, D, F, G, H, and I inhale air pressure between the range of 0 to 50 kPA. The Box plot of the feature Long-Trim Bank (Figure 4c) is also presented. This feature explains the correction value being used by the fuel control system, and it is expressed in percentages. Fuel trims are explained as a change in fuel over some time. Long trim fuel bank-1 means that the powertrain control module detects a fuel trim outside the range of specification set by the vehicle's manufacturer. There are two types of fuel trims Short Term Fuel Trim (STFT) and Long Term Fuel Trim (LTFT). The Box plot of this feature shows 10 drivers' distribution, and it can be seen that driver A has the lowest distribution, and driver

E has the highest distribution. The other drivers show the distribution between 2 to 6%. The box plot of feature friction of torque is ranged from 0-100% (Figure 4d). From the distribution, it can be seen that driver A exhibits the highest percentage as compared to other drivers. In contrast, other drivers B, C, D, E, F, G, H, I, and J show a similar kind of distribution between 10 to 15%. The range of Maximum Indicated Engine Torque is between 0-100% (Figure 4e). From the analysis of the distribution of boxes of different drivers, it can be seen that all the drivers are having a distribution range of 55 to 75%. Box plot related to feature Transmission oil temperature is shown in Figure 4f. The feature shows the temperature of oil inside the transmission. The range of this feature is between -40 to 215 Celsius degree. Driver A has the highest value 100 C, and other drivers have a distribution between ranges of 85-95 C. The only exception in this feature is the value of driver E, which has the lowest temperature.

5. Results and Evaluation

Our proposed model, ATVSA, has three main steps: driver identification, driver verification, and driver detection. For our proposal of the semi-supervised learning-based driver identification model, we have used a set of supervised and unsupervised learning algorithms. We have applied four algorithms mainly J48, Random Forest, JRip and PART for supervised learning, and K-means/Canopy clustering for unsupervised learning using the WEKA tool in order to create a pool of unique driving styles.

5.1. Driver Identification

We have chosen those algorithms for driver identification that have shown acceptable performance in previous works, as far as the accuracy of these algorithms is concerned. Driver identification training is performed every second, as the unique driving patterns are recorded every second. 10-fold cross-validation is used for the training purpose, as this technique divides the data into 10 parts, trains the model with 9 parts and 1 part is used to evaluate the model. High accuracy and generalization ability are the reasons for choosing this technique for validation purposes. Table 3 shows the accuracy of algorithms in identifying a thief driver in our experiment. It can be observed that Kappa Statistics (for inter-rater reliability for the values in the driving data) is almost the same for all algorithms. Mean Absolute Error (MAE), to calculate errors between pairs and Root Mean Squared Error (RMSE), to calculate differences between values, are higher for the Random Forest algorithm. Relative Absolute Error (RAE) for calculating the performance of the predictive model is lower for the PART algorithm, and Root Mean Squared Error (RMSE) for calculating the error rate of the regression model is lower for the J48 algorithm.

Figure 5 shows the time taken by each algorithm for training purposes using 10-fold cross-validation. We can deduce some useful information about the performance of these algorithms. We applied different algorithms and observed training time for all algorithms. We can observe that the decision tree algorithm J48 took the least training time whereas the Random Forest, a rule-based algorithm took the highest training time to train the model [38].

TABLE 3: Accuracy and Statistics of Machine Learning Algorithms

Algo.	Average Accuracy	Kappa Statistics	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
J48	99.9725	0.9997	0.0001	0.0073	6.4911	35.9954
RF	99.9629	0.9995	0.0029	0.0178	6.9001	33.9697
JRip	99.964	0.9996	0.0001	0.0081	6.5043	36.0095
PART	99.9682	0.9996	0.0001	0.0079	6.4786	36.0095

In the second part of the experiment, we have applied an unsupervised learning algorithm named K-means/Canopy clustering. We provided the driving data to the algorithm (K-means/Canopy) as an input. Using the K-means/Canopy clustering algorithm, we formed the clusters of driving data that are considered as the pool of trusted driving styles.

5.2. Driver Verification - Reconstruction of Validation Data

Reconstruction is the process for creating testing data to examine how the validation data is different from the original driving data of drivers/owners. The first stage of creating the data is to perform feature selection and feature engineering. After performing feature selection and engineering, we get the data nearest to owner data in both cases of learning. In supervised learning, we get the data that is nearest to drivers and in unsupervised learning, we obtain the data nearest to the cluster/pool of driving patterns of the owner-driver. The central values of each cluster can represent clusters created on the driving data of owners. After plotting the validation data into the same clusters, a single centroid exists in these clusters.

Error Calculation is the gap between the original driving data and reconstructed validated data. If there is a considerable gap between these types of data, there will be an error and drivers will not be classified into predefined classes.

Supervised Learning – Error Calculation: The testing is performed to examine the similarity between the original data and the validated data. If the new data samples can be classified into predefined classes, there will be no errors and drivers will be classified into authorized drivers.

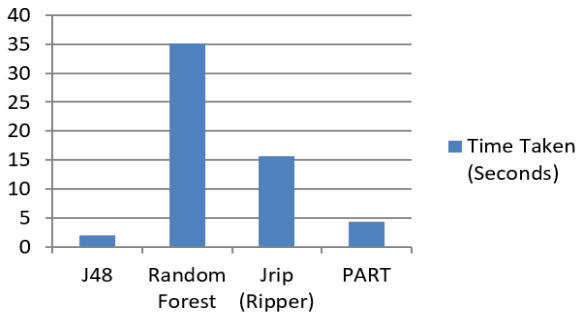
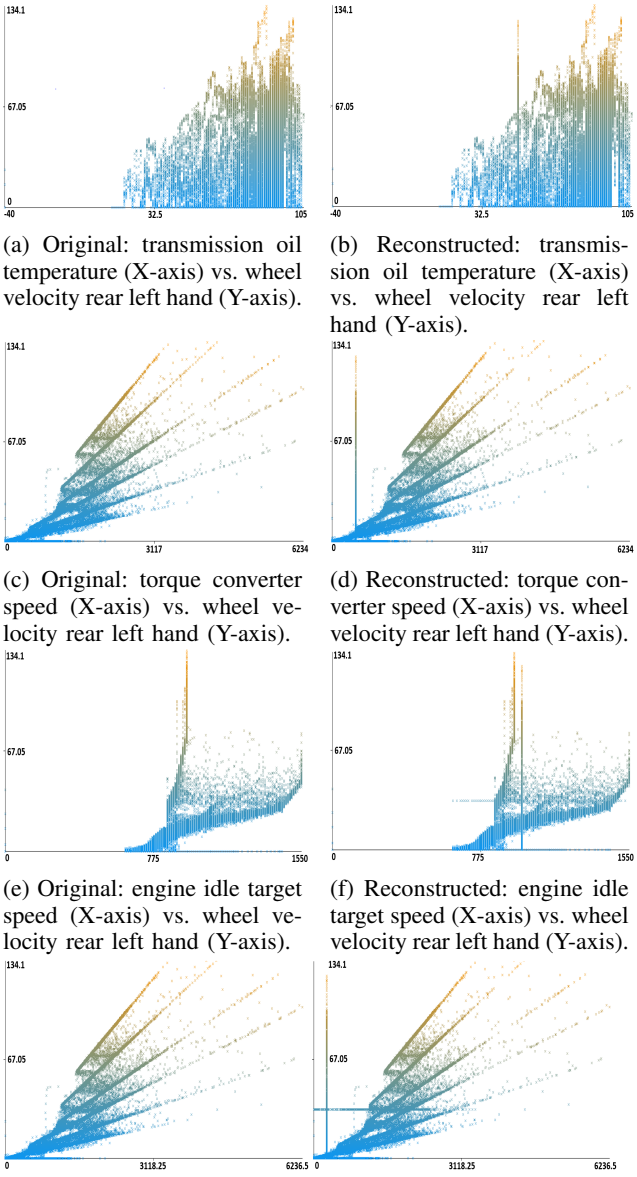


Figure 5: Execution time of algorithms on training dataset.



(a) Original: transmission oil temperature (X-axis) vs. wheel velocity rear left hand (Y-axis).

(b) Reconstructed: transmission oil temperature (X-axis) vs. wheel velocity rear left hand (Y-axis).

(c) Original: torque converter speed (X-axis) vs. wheel velocity rear left hand (Y-axis).

(d) Reconstructed: torque converter speed (X-axis) vs. wheel velocity rear left hand (Y-axis).

(e) Original: engine idle target speed (X-axis) vs. wheel velocity rear left hand (Y-axis).

(f) Reconstructed: engine idle target speed (X-axis) vs. wheel velocity rear left hand (Y-axis).

(g) Original: torque converter turbine speed (X-axis) vs. wheel velocity rear left hand (Y-axis).

(h) Reconstructed: torque converter turbine speed (X-axis) vs. wheel velocity rear left hand (Y-axis).

Figure 6: Detecting driving pattern from clustering: original (legitimate) and reconstructed (injected malicious data).

For this, we need to set up a threshold on the accuracy of classification performed by the algorithms. The threshold is the minimum percentage of accuracy attained by the algorithms. The threshold value for generating an alarm is set to be 97%.

Unsupervised Learning- Error Calculation: If the driver is the owner, the nearest driving pattern would be visible in the owner-driver data clusters created using K-means/Canopy clustering. When the search takes place for the nearest driving pattern within the pool of clusters, the distance between validation data and the nearest cluster center becomes smaller. As far as the distribution is concerned, the reconstructed data and original data will be distributed similarly. In case, the driver is a thief driver, the driving data will not be present in the clusters of the owner-driver data and the distance between validated data and the center of clusters becomes large. We set this error as detection criteria for the auto-theft cases.

We demonstrated the *reconstruction error* as a useful measure for the auto-theft detection using the K-means/Canopy clustering. To maximize the performance for theft detection, we followed the *Elbow Method* that estimates the Sum of Squared Errors (SSE) to find the optimal size of K [39]. In this technique, the distance of observations from their cluster centroids is known as the SSE. The SSE starts to decrease with the increase in the value of K. After clustering the data with optimal value K, we used the validation data to classify the owner drivers from the thief driver. We used 300 as the value of K for getting an accurate performance.

In Figure 6, we have chosen the trips undertaken by the drivers as the original driving data. To demonstrate the stolen case, we substituted the original data with fabricated driving data by repeating the specific values several times within the features that show the spike within the data clusters (Figure 6b, 6d, 6f, and 6h). We selected the features for plotting the graphs on the X and Y axis and tried to demonstrate the graphs' reconstruction error. The reconstructed data is distributed similarly to the original data. By comparing Figure 6a with 6b, Figure 6c with 6d, Figure 6e with 6f, and Figure 6g with 6h, it can be observed that reconstruction errors are increased with the introduction of new data for transmission oil temperature, torque converter speed, engine idle target speed, and torque converter turbine speed. These algorithms are able to detect when a thief and not the owner drives a vehicle.

5.3. Driver Detection

This part of the work discusses the third step, i.e., driver detection. This step aims first to detect the driver. If a thief drives the vehicle, the proposed model generates an alarm to the owner-driver that the car has been stolen, as the applied algorithms can detect high spikes and other noticeable changes in the dataset.

We evaluated our model by introducing a thief driver within the original data. The primary aim is to determine the accuracy by which the algorithms do not classify the drivers who were not part of the training dataset. We introduced a thief driver with different driving characteristics. In this experiment, we injected around 5,000 malicious values of 15 selected features of driver A (i.e., first driver). Now, using training data, we examined the instances that were correctly and incorrectly classified by the algorithms. We found that malicious values are incorrectly classified and the accuracy of the classifier is also dropping below the threshold value of 97%. Surprisingly, this time, Random Forest performed sightly better than J48, in terms of correctly classified instances and root relative squared error. The PART algorithm was better in relative absolute error, while JRip (Ripper) was found as the worst performer compared to other algorithms.

We tested the model with a varied dataset size for training and testing. We have established a concept called window size in our approach that is used as a notification time for owners in case of theft. For fast detection, the window size is kept small, and for reliable accuracy, it

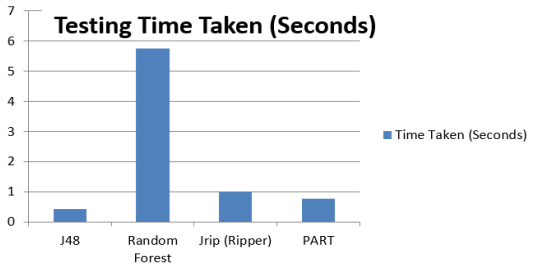


Figure 7: Execution time of algorithms on testing dataset.

should be kept long enough. The sliding window can also be used as the measure of receiving the optimal accuracy of classifiers. During the work, we also noted that the accuracy of algorithms is directly proportional to the sliding window size. To achieve the high accuracy of algorithms, we need to increase the sliding window size to around 80-100 seconds. However, with high accuracy, there is also a drawback to the proposed model that it will be notifying the owner about the theft with some delay. We have also analyzed the time taken by the classifiers on the testing dataset. Figure 7 shows the time taken by the classifiers to achieve an accuracy of 99%. It can be observed that the J48 algorithm took the lowest time for testing as compared to other algorithms. Note that the training time for the J48 algorithm was also the lowest. The efficiency of J48 is also demonstrated by Figure 8 that shows the execution time on the same accuracy for other algorithms. However, root relative squared error is low in the Random Forest algorithm's execution, but shows high relative absolute error, as related Kappa statistics are shown in Figure 9.

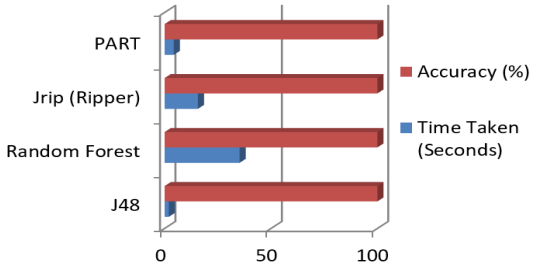


Figure 8: Time taken by the classifiers with high accuracy.

5.4. Discussion: Why Semi-supervised Learning More Suitable?

The purpose of using semi-supervised learning is to improve the overall accuracy of the approach using supervised and unsupervised algorithms. Now the question is, why use this approach, as we can still get high accuracy by using only the supervised or unsupervised learning approach in the work? Actually, using a semi-supervised learning approach, we solve a supervised learning approach using labeled data augmented by unlabeled data. The number of unlabeled or partially labeled instances is often larger than the number of labeled instances since the former is less expensive and easier to obtain. Therefore, our goal is to overcome one of the problems of supervised learning, i.e., having not enough labeled data. Adding

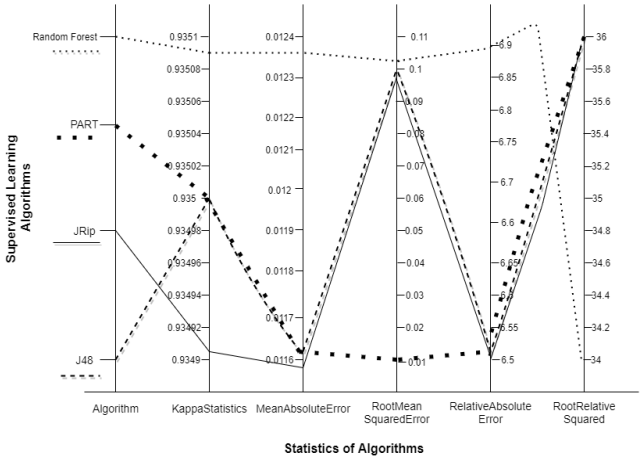


Figure 9: Comparison of algorithms' statistics.

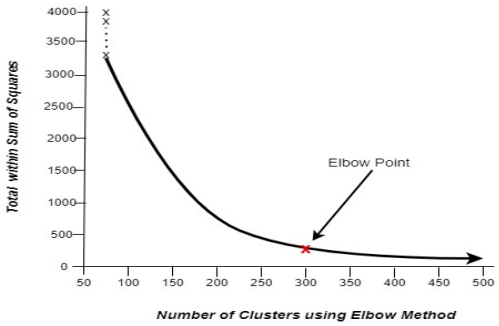


Figure 10: Sum of squared errors for each value of K.

cheap and abundant unlabeled data, we hope to build a better model than using supervised learning alone. In a real-world scenario, we receive a mix of data having labels and no labels attached to it. Having a hybrid approach can cater to both types of data (label and not labeled) that is more suitable under our proposed model, ATVSA, for driver identification.

Figure 10 shows the Sum of Squared Errors (SSE) for each value of K. The line in the graph looks like an arm and the elbow on the arm is the value of K. Our aim is to keep the SSE as small as possible. Our proposed ATVSA model uses supervised and unsupervised learning algorithms, we have compared the performance of our model with the existing works. Figure 11 shows a comparison of the accuracy of supervised learning algorithms. The accuracy of supervised learning algorithms used in existing works Zhang et al. [32], Meng et al. [23], Nishiwaki et al. [27], Choi et al. [8], Enev et al. [29], Wahab et al. [30], Kedar-Dongarkar et al. [31], and Our ATVSA is 85%, 99%, 76.80%, 25%, 87%, 99%, 77%, and 99.90%, respectively.

6. Conclusion

In this work, we proposed a vehicle driving profiling model for theft detection, i.e., ATVSA, with comparative results of supervised learning with the pool of owner driver's data. We use statistics to better understand the accuracy and performance of these algorithms by classifying drivers on their unique driving patterns. In this work, CAN data extraction and feature preprocessing are used as

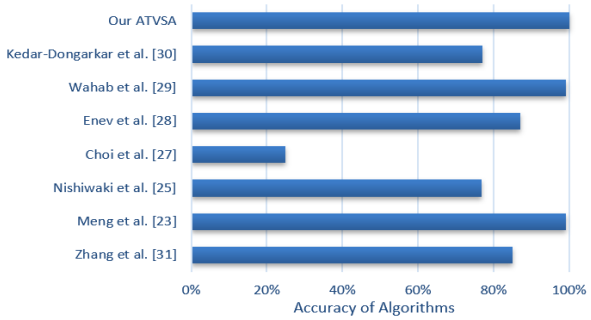


Figure 11: Comparing existing works - supervised algorithms.

a form of data analysis and augmentation. To identify the user on the basis of his driving habits, algorithms like the Decision Tree, Random Forest and K-nearest neighbors have been applied. Multiple models were laid out and trained and the best results were obtained. Also, more precise identification of drivers' profiles is established. Some of the driver's profiles were stored in the system as authentic users, and if the driving pattern deviates from the ones recognized, an alert is sent to the owner of the vehicle. The alert notifies the owner about the presence of an unknown driver.

We have shown that semi-supervised learning can be used to detect auto-theft cases. Furthermore, we have used both, supervised and unsupervised learning algorithms to show how successful and accurate our ATVSA model approach is in detecting the auto-theft cases. It can be observed from the results that the J48 algorithm outperformed in both, training and testing time. Moreover, under unsupervised learning, a clustering algorithm (K-means/Canopy) is used to cluster the driving data into a pool of trusted driving styles. We have also demonstrated the optimal use of window size that plays a vital role in increasing the classifiers' accuracy. We fixed the sliding window size to 50 seconds for trading the time with accuracy.

References

[1] Abdellah Mekki, Afaf Bouhoute, and Ismail Berrada, "Improving driver identification for next-generation of in-vehicle software systems," *IEEE Trans. on Veh. Tech.*, 68, 8, Aug. 2019, pp. 7406-7415.

[2] Chandrasekar Ravi, Anmol Tigga, G Thippa Reddy, Saqib Hakak, and Mamoun Alazab, "Driver Identification Using Optimized Deep Learning Model in Smart Transportation," *ACM Trans. Internet Technol.*, July 2020.

[3] Hashim Abu-gellban, Long Nguyen, Mahdi Moghadasi, Zhenhe Pan, and Fang Jin, "LiveDI: An Anti-theft Model Based on Driving Behavior," *ACM Workshop on Information Hiding and Multimedia Security*,2020, pp. 67–72.

[4] Khattab M. Ali Alheeti, Anna Gruebler, and Klaus D. McDonald-Maier, "An intrusion detection system against malicious attacks on the communication network of driverless cars," *IEEE CCNC*, 2015, pp. 916-921.

[5] Adrian Taylor, Sylvain Leblanc, and Nathalie Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," *IEEE Inter. Conf. on Data Science and Advanced Analytics (DSAA)*, Montreal, Canada, 2016, pp. 130-139.

[6] Ferhat Attal, Abderrahmane Boubezoul, Allou Samé, and Stéphane Espié, "Powered two-wheelers critical events detection and recognition using data-driven approaches," *IEEE Trans. on Intelligent Transportation Systems* 19, 12, Dec. 2018, pp. 4011-4022.

[7] Minh Ly, Sujitha Martin, and Mohan Trivedi, "Driver classification and driving style recognition using inertial sensors," *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, 2013, pp. 1040-1045.

[8] SangJo Choi, JeongHee Kim, and John Hansen, "Analysis and classification of driver behavior using in-vehicle can-bus information," *DSP for in-vehicle and mobile systems Workshop*, 2017, pp. 17-19.

[9] D. Clark, Number of motor vehicle theft offences in England & Wales 2002-2019, 2019. https://www-statista-com.abc.cardiff.ac.uk/statistics/303551/motor-vehicle-theft-in-england-and-wales/.

[10] Omar Al-Jarrah, Carsten Maple, Mehrdad Dianati, David Oxtoby, and Alex Mouzakitis, "Intrusion detection systems for intra-vehicle networks: A review," *IEEE Access* 7, 2019, pp. 21266-21289.

[11] Jhon Evans, 2020. Why are car thefts still on the rise? 2020. https://www.autocar.co.uk/car-news/features/why-are-car-thefts-still-rise.

[12] Jie Chen, Zhong Cheng Wu, Jun Zhang, and Song Chen, "Driver identification based on hidden feature extraction by using deep learning," *3rd Infor. Tech., Networking, Electronic and Automation Control Conf.*, Chengdu, China, 2019, pp. 1765-1768.

[13] Zoran Constantinescu, Cristian Marinoiu, and Monica Vladoiu, "Driving style analysis using data mining techniques," *Inter. Journal. of Computers, Communications & Control*, 5, 2010, pp. 654-663.

[14] Fabio Martinelli, Francesco Mercaldo, Albina Orlando, Vittoria Nardone, Antonella Santone, and Arun K. Sangaiah, "Human behavior characterization for driving style recognition in vehicle system," *Computers and Electrical Engineering*, 83, 2020, p. 102504.

[15] Byung Il Kwak, JiYoung Woo, and Huy Kang Kim, "Know your master: driver profiling-based anti-theft method," *Annual Conference on Privacy, Security and Trust (PST)*, Auckland, New Zealand, 2016, pp. 211-218.

[16] David Hallac, Abhijit Sharang, Rainer Stahlmann, Andreas Lamprecht, Markus Huber, Martin Roehder, and Jure Leskovec, "Driver identification using automobile sensor data from a single turn," *Inter. Conf. on Intelligent Transportation Systems*, Rio de Janeiro, Brazil, 2016, pp. 953-958.

[17] Sotirios Katsikeas, Pontus Johnson, and Robert Lagerstr, "Probabilistic modeling and simulation of vehicular cyber attacks: an application of the meta attack language," *International Conference on Information Systems Security and Privacy*, 2019, pp. 175-182.

[18] Yuguang Huang and Lei Li, "Naive Bayes classification algorithm based on small sample set," *Inter. Conf. on Cloud Computing and Intelligence Systems*, Beijing, China, 2011, pp. 34-39.

[19] Anjali Jivani, "Novel k Nearest Neighbor Algorithm," *Inter. Conf. on Computer Comm. and Informatics*, Coimbatore, 2013, pp. 1-4.

[20] T. Feuring, "Learning in fuzzy neural networks," *Inter. Conf. on Neural Networks (ICNN)*, Washington, USA, 1996, pp. 1061-1066.

[21] Daniel Forster, Robert B. Inderka, and Frank Gauterin, "Data-driven identification of characteristic real-driving cycles based on k-means clustering and mixed-integer optimization," *IEEE Transactions on Vehicular Technology* 69, 3, 2020, pp. 2398-2410.

[22] Xingjian Zhang, Xiaohua Zhao, and Jian Rong, "A study of individual characteristics of driving behavior based on hidden markov model," *Sensors and Transaction* 167, 3, 2014, pp. 194-202.

[23] Xiaoning Meng, ka keung and Yangsheng Xu. ' ' Driving Behavior Recognition Based on Hidden Markov Models," *IEEE Inter. Conf. on Robotics and Biomimetics*, 2006, pp. 274-279

[24] S. fasiuddin, S. Omer, K. Sohelrana, A. Tamkeen and M. A. Rasheed, "Real Time Application of Vehicle Anti Theft Detection and Protection with Shock Using Facial Recognition and IoT Notification," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 1039-1044.

[25] T. J. Claude, I. Viviane, I. J. Paul and M. Didacienne, "Development of Security Starting System for Vehicles Based on IoT," 2021 Intern. Conf. on Information Technology (ICIT), 2021, pp. 505-510.

[26] A. M. Ali, H. M. Awad and I. K. Abdalgader, "Authenticated Access Control for Vehicle Ignition System by Driver's License and Fingerprint Technology," 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2021, pp. 1-6.

[27] Nishiwaki, Yoshihiro, Koji Ozawa, Toshihiro Wakita, Chiyomi Miyajima, and Kazuya Takeda, "Driver identification based on spectral analysis of driving behavioral signals," *In Advances for In-Vehicle and Mobile Systems* Bostan, MA, 2010, pp.25-34.

[28] Jun Zhang, Zhong Cheng Wu, Jie Chen, and Liu Liu, "A deep learning framework for driving behavior identification on in-vehicle CAN-BUS sensor data," *Sensors*, 19, 6, 2019, p. 1356.

[29] Miro Enev, Alex Takakuwa, Karl Koscher, and Tadayoshi Kohno, "Automobile driver fingerprinting," In *Proceedings of Privacy Enhancing Technologies*. Sciendo, 2016, 35-40.

[30] Abdul Wahab, Chai Quek, Chin Tan, and Kazuya Taked, "Driving profile modelling and recognition based on soft Computing approach," *IEEE Trans. on Neural Networks* 20, 4, 2009, pp. 563-582.

[31] Kedar-Dongarkar, Gurunath, and Manohar Das, "Driver classification for optimization of energy usage in a vehicle." *Procedia Computer Science*, 8,2012, 388-393.

[32] Adi Karahasanovic, Pierre Kleberger, and Magnus Almgren, "Adapting threat modelling methods for the automotive industry," *15th ESCAR Conference*, Berlin, Germany, 2017, pp. 1-10.

[33] Zoran Constantinescu, Cristian Marinoiu, and Monica Vladoiu, "Driving style analysis using data mining techniques," *Inter. J. of Computers, Communications & Control*, 2010, 5, 2010, pp. 654-663.

[34] Bryan Higgs and Montasir Abbas, "Segmentation and clustering of car-following behavior: recognition of driving patterns," *IEEE Trans. on Intelligent Transportation Systems* 16, 1, 2015, pp. 81-90.

[35] Xiaoning Meng, Ka Lee, and Yangsheng Xu, "Human driving behavior recognition based on hidden markov models," *Inter. Conf. Robotics and Biomimetics*, Kunming, China, 2006, pp. 274-279.

[36] HCRL, Driving Dataset. https://ocslab.hksecurity.net/Datasets/driving-dataset

[37] Syed Pasha and E. Syed Mohamed, "Ensemble gain ratio feature selection (EGFS) model with machine learning and data mining algorithms for disease risk prediction," *Inter. Conf. on Inventive Computation Technologies*, Coimbatore, India, 2020, pp. 590-596.

[38] Ahmed Ahmim, Leandros Maglaras, Mohamed A. Ferrag, Makhlouf Derdour1, and Helge Janicke, "A Novel hierarchical intrusion detection system based on decision tree and rules-based models," *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Santorini, Greece, 2019, pp. 228-233.

[39] Purnima Bholowalia and Arvind Kumar, "EBK-means: A clustering technique based on elbow method and k-means in WSN," *International Journal of Computer Applications*, 105, 9, 2014.