Discrete Wavelet Transform for Generative Adversarial Network to Identify Drivers Using Gyroscope and Accelerometer Sensors
Ruhallah Ahmadian, Mehdi Ghatee, Johan Wahlström

Abstract—Driver identification is an important research area in intelligent transportation systems, with applications in commercial freight transport and usage-based insurance. One way to perform the identification is to use smartphones as sensor devices. By extracting features from smartphone-embedded sensors, various machine learning methods can identify the driver. The identification becomes particularly challenging when the number of drivers increases. In this situation, there is often not enough data for successful driver identification. This paper uses a Generative Adversarial Network (GAN) for data augmentation to solve the problem of lacking data. Since GAN diversifies the drivers' data, it extends the applicability of the driver identification. Although GANs are commonly used in image processing for image augmentation, their use for driving signal augmentation is novel. Our experiments prove their utility in generating driving signals emanating from the Discrete Wavelet Transform (DWT) on smartphones' accelerometer and gyroscope signals. After collecting the augmented data, their histograms along the overlapped windows are fed to machine learning methods covered by a Stacked Generalization Method (SGM). The presented hybrid GAN-SGM approach identifies drivers with 97% accuracy, 98% precision, 97% recall, and 97% F1-measure that outperforms standard machine learning methods that process features extracted by the statistical, spectral, and temporal approaches.

Index Terms—Data Augmentation, Driver Identification, Generative Adversarial Network, Ensemble Learning, Smartphone Sensors, Stacking, Smartphone Sensors.

I. INTRODUCTION

Artificial intelligence and data mining are two essential paradigms in developing future transportation systems. Thanks to the large amounts of traffic data collected from sensors in automobiles, telecommunications antennas, and smartphones, these paradigms have rapidly transformed the transportation sector. As an instance, a driver identification problem classifies the drivers based on location or behavioral characteristics. It is applicable in such diverse areas as freight transportation, driver control, anti-theft systems, and usage-based insurance systems [1], [2].

Previously, drivers were identified using simple devices such as ID cards. Following this, more advanced methods such as fingerprint, face recognition, and finger vein pattern recognition have been applied. However, these methods often violate the driver's privacy or enable the drivers to cheat. Nowadays, the modern driver identification systems collect data from in-vehicle sensors [3], GPS [4], inertial sensors [5], or their combination. The related literature have used the following techniques to classify the drivers:

- Statistical methods such as Gaussian Mixture Models [6] and Hidden Markov Models [7].
- Unsupervised learning methods such as Generative Adversarial Network (GAN) [1].
- Supervised learning methods based on the user-defined features, such as Linear Discriminant Analysis [8] and Extreme Learning Machines [9].
- Deep networks based on the hidden features such as Gated Recurrent Unit (GRU) [5].
- Ensemble methods such as gradient tree boosting [10], random forest [11], extra trees [12], and Stacked Generalization Method (SGM) [13].

Since the CAN-bus is available in most modern cars, many researchers used their data for driver identification; see Table I. However, some vehicles are still not compatible with this standard, and it is a motivation to use accelerometers and gyroscopes for driving identification. Smartphones and OBD-II
TABLE I: Selected driver identification studies

| Ref  | Sources          | Signals                                      | Features                              | Models                      | Drivers | Accuracy |
|------|------------------|----------------------------------------------|---------------------------------------|-----------------------------|---------|----------|
| [14] | CAN-bus          | 48 ECU signals                               | Statistical, Descriptive, Frequency   | SVM, Random Forest, Naive Bayes, KNN | 15      | 100%     |
| [15] | Smartphone       | Acceleration                                 | Statistical                           | PCA                         | 5       | 60%-100% |
| [5]  | Smartphone       | Acceleration, Angular Velocity               | CNN feature maps                      | Resnet50 + GRU              | 25      | 71.89%   |
| [7]  | Event Simulator  | Acceleration, Brake, Steering Wheel          | Driving maneuvers such as deceleration, change lane | Hidden Markov Model         | 20      | 85%      |

devices support these sensors. Accelerometers measure a specific force, whereas gyroscopes measure angular velocity. [16] identified drivers by acceleration, angular velocity, and lane changing features extracted from GPS data. [15] used PCA on smartphone acceleration data to identify drivers, although its patterns were similar for some drivers. The histograms of acceleration data were also fed to a Multi-Layer Perceptron (MLP) by [17]. [5] transformed the acceleration signal and the angular velocity to a spectrogram and then processed them by Convolutional Neural Networks (CNN) and GRU. [7] used a Hidden Markov Method (HMM) on simulated driving data. Ensemble methods are also applicable to smartphone data. For example, [18] applied an ensemble of CNN, Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) for driving identification. Table I presents details of these studies and their performances. Despite the broad literature on driver identification, some challenges have remained:

- The performance of the applied machine learning models is sensitive to the examined scenario.
- The proposed models are usually overfitted to the training data, and their performance is poor on new data.
- As the number of drivers increases, the driver identification accuracy decreases.

To solve these challenges:

1) We consider different trained machine learning methods in the SGM ensemble method to get their specialties in an integrated system.
2) By augmenting the driving data, we overcome the overfitting problem.
3) The original driving data and the generated data are used in the training process of SGM to improve the generalization performance.

The proposed hybrid method of GAN and SGM contains the following contributions:

- GAN has been widely used for image augmentation to improve image classification. Besides, [1] used the GAN’s discriminator part to identify automobile theft. However, GAN usage to augment driving signal data is the novelty of this paper and controls the overfitting. This approach causes stability in the results, and our system’s accuracy becomes independent of the examined drivers.
- To train the GAN model, DWT of 6 signals of three-dimensional accelerometer and gyroscope sensors data are fed to a specialized GAN to generate some augmented drivers’ data.
- In the feature extraction phase, we apply histograms of acceleration and gyroscope signals on overlapped windows, while [17] used only acceleration data. Although the GAN model works on DWT of driving data and yields promising results, the histogram feature overcomes DWT, spectral, temporal, and other statistical features for the driver identification phase.
- The classification is done using the SGM on the extracted features. Likewise, [13] has used SGM on in-vehicle data and identified the drivers with 88% accuracy. Our SGM on smartphone data improves this accuracy up to 97%.

In what follows, Section II presents the driver identification system. The experiments and sensitivity analysis results are given in Section III. The final section ends the paper with a brief conclusion.

II. PROPOSED SYSTEM

A novel architecture for driver identification is presented in Fig. 1. This architecture covers data pre-processing, data augmentation, feature extraction, and driver identification. In the following subsections, we present details of the individual modules.

A. Data pre-processing module

This module collects the acceleration, angular velocity, and magnetic field signals and produces segmented data by pursuing six steps:

1) The sensors’ data should be aligned with vehicle axes as the cellphone may be located in various positions. We use the dataset [19] that aligned signals based on the magnetic field.
2) After reorientation, only acceleration and angular velocity are used for data cleaning. The outliers are clipped, and then the missing values are replaced by their average.
3) We then remove the motionless sampling instances, by testing \( \sum_{t=1}^{t+6f_s} \sqrt{\alpha_x^2 + \alpha_y^2 + \alpha_z^2} \leq \theta \), where \( \theta = 0.5 \) and the frequency is \( f_s = 2 \) Hz.
4) The data is normalized using the Gaussian transformation \( x' = \frac{x-\bar{x}}{\sigma} \), where \( \bar{x} \) and \( \sigma \) denote average and standard deviation, respectively.
5) The data is split into segments using overlapped windows. The parameters of this step are window size and overlap percentage.

We have GAN-training and SGM-training phases. For the GAN-training, the existing dataset is split into training and validation parts. When GAN learns the training data, we save it and use it for data generation. The combination of generated
Fig. 1: The architecture of the proposed hybrid model for driver identification consists of two separate stages. First, a GAN model is trained to data augmentation, and then a SGM is trained to identify drivers.

data and the training data is guided to segmentation step. For SGM-training, the data is directly guided to the feature engineering module.

On the other hand, overfitting is a common problem in machine learning, creating significant challenges for various driver identification systems. One of the best solutions for overfitting is data augmentation [20]. Generative Adversarial Methods [21] can augment data to enhance the generalization performance. GANs indirectly learn the distribution of data $p_{data}$ from a sequence of samples $x^{(1)}, x^{(2)},..., x^{(n)}$ to produce new samples from the learned distribution with the same statistics as the training set.

GAN includes two neural networks that are trained concurrently. The first network is known as the discriminator ($D$). This network takes sample $x$ as input and returns $D(x)$, which is the probability that $x$ is an actual sample. The second network is called the generator ($G$). This network synthesizes samples to make $D$ believe they are actual samples. $G$ takes some random input $z^{(1)}, z^{(2)},..., z^{(n)}$ usually from a uniform or Gaussian distribution, and maps $G(z)$ to the data space of the distribution $p_{data}$. Thus, the aim of $G$ is to ensure that $p_{G} = p_{data}$. GANs are trained by optimizing the following objective function as a minimax game [21]:

$$\min_{G} \max_{D} \mathbb{E}_{x \sim p_{data}} \log D(x) + \mathbb{E}_{z \sim p_{z}} \log D(1 - G(z)), \quad (1)$$

where $x \sim p_{data}$ and $z \sim p_{z}$ indicate that the distributions of $x$ and $z$ are $p_{data}$ and $p_{z}$. $\mathbb{E}$ is the mathematical expectation. The discriminator maximizes the logarithm of $D(x)$ for $x \sim p_{data}$ and minimizes for $x \not\sim p_{data}$. The generator fools $D$ by creating samples that $D(G(z)) \sim p_{data}$. In other words, the generator maximizes $D(G(z))$, which is equivalent to minimizing $1 - D(G(z))$. [21] showed that GAN generates new samples from a standard data distribution when its objective function converges to $-\log(4)$.

1) Discrete Wavelet Transform (DWT): On accelerometer and gyroscope signals, here denoted $s(n)$, we use DWT. Any signal $s(n)$ of discrete variables $n = 0, 1, ..., M - 1$, can be transformed to the following components [22]:

$$s(n) = \frac{1}{\sqrt{M}} \left( \sum_{k} W_{\phi}(0,k) \phi_{0,k}(n) + \sum_{j=0}^{J-1} \sum_{k} W_{\psi}(j,k) \psi_{j,k}(n) \right)$$

where $\phi_{0,k}(n)$ and $\psi_{j,k}(n)$ are mother wavelet functions, $W_{\phi}(0,k)$ and $W_{\psi}(j,k)$ are approximation and detail coefficients, where $J = \log_{2} M$, $j = 0, 1, 2, ..., J - 1$ and $k = 0, 1, 2, ..., 2^{j} - 1$. Besides, by Inverse Discrete Wavelet Transform (IDWT), one can directly construct the original signal from the approximation and detail series. Now, following [19], we use the Daubechies family of mother wavelet (DB2). For each window of accelerometer and gyroscope signals in 3-dimensions, we derive approximation and detail coefficients vectors $W_{\phi}$ and $W_{\psi}$. We have six vectors in each segment as we have six axes in the input, and because each segment is decomposed one level so that we have twelve vectors in each segment. These 12 vectors are used as the features of the overlapped windows for the GAN training phase; see Fig. 2 that visualizes this process.

Fig. 2: The extracted features by DWT from accelerometer (green rectangles) and gyroscope (orange rectangles) signals for overlapped windows.

2) GAN Training: We examine several GAN models, including vanilla GAN [21], and deep convolutional GAN [23].
Both are implemented under conditional GAN [24]. Their accuracies for the driver identification system are compared when they augment the dataset. A GAN that leads to the highest accuracy is selected for the proposed system. The experiments showed that:

- Augmentation did not improve driver identification accuracy when the GAN model contains too various layers or only dense layers.
- Generated data by deep convolutional GAN, shown in Fig. 3, was significantly robust and advanced the accuracy. At least 2,000 iterations were needed to train this GAN model.

To process the time-streaming signals, some windowing functions, including tumbling, hopping, sliding, session, and snapshot windows, can be used. In our GAN models, we used sliding windows for input signals and tumbling windows for output signals. We considered 75% overlap for sliding windows. Tumbling windows are also disjoint. The lengths of both windows were equal. Denote $TW = 10$ seconds as the length of tumbling windows. For generating $w = 15$ minutes of driving data, the GAN model generated $n = w/TW = 90$ tumbling windows. These tumbling windows were saved in a tensor with size $[n, 12, \lceil TW/2 \rceil + 1]$ in DB2 wavelet space. $\lceil TW/2 \rceil + 1$ is the number of approximation and detail coefficients of each tumbling window. By IDWT and combining the approximation and detail coefficients, these tumbling windows reconstruct back into the original signal and then they were reshaped to $[n \times TW, 6]$ to meet the training data standard.

Fig. 3: Deep convolutional GAN architecture for generating driving data.

B. Feature engineering module

Different statistical, temporal and spectral features are extracted for each window data by a library given in [25]. These features are commonly applied in signal processing, such as speech processing, electrocardiogram analysis, and driver behavior evaluations. Previous driver identification researches also used some of these features. For example, [26] used statistical features, [17] extracted histograms from acceleration data, [9] used temporal features, and [11] utilized spectral features.

However, considering all features cannot necessarily improve the classification performance. The most relevant features to the labels with the least redundancy are more informative for shallow and deep learning models. Thus, a critical part of machine learning problems is feature selection. We used both filter and wrapper methods for feature selection. The filter method is independent of machine learning algorithms, and the features are sorted based on their scores. Here, we use Pearson’s correlation coefficient to score the features and remove the unrelated ones whose correlations are more diminutive than 95%. Then, we use a wrapper method that selects different subsets of the remaining features from the filter method and trains the driver identification model using these features. The best accuracy of the model leads to the best set of features. We derived that histogram features of accelerometer and gyroscope data are the best features for driver identification.

We use mostly six windows in each extraction cycle and extract their features separately. They store together in a single row in a new dataset. A grid search on histogram features demonstrates that the optimal number of bins is 100. Also, reducing the histograms’ scope to 95% shows the best robustness in the final model. Finally, we normalize the remaining features on their $L_2$-norms.

C. Identification module

In driver identification by classification methods, every driver is considered one class, and the model tries to recognize this class based on the extracted features. Each model first learns training features, and then its performance is evaluated on the testing data. Here, a wide range of machine learning algorithms, including Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Gaussian Naive Bayesian (GNB), and ensemble algorithms are compared. Ensemble methods such as Boosting, Bagging, Adaboost (AB), Gradient Boosting (GB), Random Forest (RF), Extra Tree (ET), Bootstrap aggregating (BG), Stacked Generalization Method (SGM), and Voting (VT) that combine multiple learning models, are also compared. Boosting and Bagging improve stability and accuracy and also reduce variance. AB and GB are examples of boosting algorithms, and RF and ET are two kinds of Bagging. Also, two bagging classifiers consisting of SVM and MLP are compared. SGM and Voting classifiers are wrappers, and they can be trained in parallel to take advantage of various algorithms. The following combinations are used in staking and majority voting:

1. Voting: \{SVM, MLP, KNN, RF\}
2. SGM: \{SVM, MLP, KNN, RF\} $\rightarrow$ LR

Based on the values of a set of predictor variables, LR predicts the presence of an outcome. The presence probability of an outcome assuming dependent variables $x_1, ..., x_n$, is given by $p = 1/(1 + \exp(-(b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n)))$, where $b_0$ is the intercept of the model, and $b_i$ (i.e., $0, 1, 2, ..., n$) are the slope coefficients. In our model, $x_i$ ($i = 0, 1, 2, ..., n$) are the outcomes of SVM, MLP, KNN, RF. By minimizing the negative logarithm of maximum likelihood estimation, we obtain the optimal $b_i$ parameters. To this aim, the gradient descent algorithm under one-vs-all is used. The results are presented in the next section.
III. Experimental Results

This section evaluates the proposed system on the data collected by [19] using different smartphones with 2 Hz sampling rate from 10 drivers. The average driving time for all drivers is 7 hours and 45 minutes. The data includes IMU and GPS data. [19] used accelerometer and magnetometer measurements for data reorientation. We use their results and only use the accelerometer and angular velocity measurements for the driver identification model to preserve privacy. To implement the algorithms, we used Scikit-learn version 0.22, Tensorflow version 1.13.2, and Keras version 2.3.0.

A. Sensitivity analysis on involved axis

The accelerometer and gyroscope sensors have three axes. We test all of their combinations to find the best driver identification results. Simply speaking, the x-axis represents the lateral axis and indicates transverse maneuvers such as changing lanes or turning. Similarly, the y-axis represents the longitudinal axis and suggests moving forward and backward maneuvers, such as acceleration or deceleration. The z-axis presents the vertical axis and shows road inclination and road obstacles, such as going over bridges or speed bumpers. Fig. 4 shows some of the best combinations of the involved axes for driver identification. This figure shows that the y-axis provides the most informative results. Also, it reveals that the use of both gyroscopes and accelerometers significantly increases accuracy in every situation. This result contradicts the results of [17], which recommended using only the accelerometer signal for 10 hours to identify a driver. Instead, we found that gyroscope and accelerometer data for 2 hours and 10 minutes per driver are sufficient to solve the same problem. Also, unlike [17] that suggested avoiding the z-axis, our results show that the highest performance achieved using data from all axes.

B. Sensitivity analysis on window properties

Finding suitable window size and overlap percentage is essential for driver identification problems. Fig. 5 shows the accuracy results on 10 drivers. According to this chart, the highest accuracy is achieved for 20-minute window size and 75% overlap. In this case, a driver can be identified every 11 minutes and 15 seconds. With the same window size and 75% overlap, a driver can be identified with 91.5% accuracy every 3 minutes and 45 seconds. The improvement by extending the overlap indicates that there is often critical information at the windows’ edges.

C. Analysis on features

As correlated variables gain no additional information, they are genuinely redundant. Besides, domains with large numbers of input variables suffer from the curse of dimensionality, and multivariate methods may overfit the data. In this research, two steps are used for feature selection. One is a fixed threshold based on correlation that removes redundant features whose correlations are less than 95%. In the second step, the remaining features are compared based on their classification accuracy. Fig. 6 shows accuracy results from the experiment on ten drivers with 15 minutes window size and 75% overlap. Based on t-SNE method [27], Fig. 7 shows different drivers with different colors. It confirms that the drivers are separable based on histograms, statistical features, and temporal features. Conversely, spectral features cannot distinguish drivers. Using all features also provides unsatisfactory results, and the model’s variance is high. The driver identification accuracy for all features is 73.1%. The best feature is the histogram that leads to 91.5% accuracy.

D. Performance of identification algorithms

The most important part of driver identification is classification. In this part, we use the histogram features of all axes of the accelerometer and gyroscope. The window size and overlap
Fig. 7: Comparison between different sets of features by t-SNE visualization for 5 experiments (colors refer to the different drivers).

Fig. 8: Comparison between the accuracies of the different classifiers for driver identification.

Fig. 9: Comparison between the accuracies of the different classifiers for different numbers of drivers.

percentage are 15 minutes and 75%. Fig. 8 illustrates the best results of different classification algorithms to recognize ten drivers. The best results are shown in a light color.

Besides, Fig. 9 compares the best algorithms to recognize the different numbers of drivers. The system accuracy is 100% for 4 and 6 drivers, 95% for eight drivers, and 91.5% for ten drivers. The best non-ensemble algorithms are MLP, KNN, and SVM. Also, SGM and Voting have the best accuracy among ensemble methods, while the Bagging method does not perform well.

Fig. 10: The performance of the discriminator and the efficiency of the generated driving data through 1.6k iterations.

E. Performance of data augmentation

To investigate the effect of data augmentation in the driver identification problem, Fig. 10 illustrates the discriminator’s results where the deep convolutional GAN generated driving data for ten drivers with window size $w = 15$ minutes. Based on the Subsection II-A.2, we used tumbling windows with $TW = 10$ seconds. The discriminator’s accuracy converged to 40%, and its loss was less than 0.7. The generator loss also converged to 0.7. To verify these results, we first got the average on the $n$ tumbling windows for each driver $d$ and each axis $s$. Denote these averages with $\bar{A}_{d,s}(t)$ and $\bar{G}_{d,s}(t)$ for the actual and generated signals over time step $t$. Then, we use the following distance to measure the dissimilarity for each tumbling window:

$$S(A, G) = \sum_{d \in \text{drivers}} \sum_{s \in \text{axes}} \int_1^{TW} |\bar{A}_{d,s}(t) - \bar{G}_{d,s}(t)| dt \quad (2)$$

The integral calculates the area between the mentioned curves. By comparing 20000 tumbling windows of driving data with the training data and testing data, the dissimilarities were 16.027 and 18.54, respectively. Thus, the quality of generated data is acceptable. Fig. 10 shows the results of dissimilarity through 1.6k iterations. As one can see, the dissimilarity converged when the number of iterations grows.

Also, by using t-SNE projection, as shown in Fig. 11, the generated data effectively discriminate the different drivers presented with the different colors for 15 minutes of data augmentation. Their effects decrease when the augmentation size becomes 120 minutes. Thus, GAN’s generated data produces diverse driving data and improves the driver identification methods’ generalization.

F. Performance of Hybrid Model of GAN and SGM

Although data augmentation is an effective method to control overfitting, it is necessary to design its implementation manually. In [20], AutoAugment procedure has been proposed to search the best policy for data augmentation. Instead of this procedure, we compared the performance of the hybrid methods of GAN and most successful classifiers of Fig. 8,
Fig. 11: The results of t-SNE projection on the training data together augmented driving data, when the size of augmentation data varies from 0 to 120 minutes (colors refer to the different drivers).

(a) 0 min per driver  (b) 15 min per driver  (c) 30 min per driver
(d) 60 min per driver  (e) 120 min per driver

Fig. 12: Comparison between the accuracy of different learning algorithms on augmented data with different augmentation sizes (in minutes).

TABLE II: The best results of hybrid models of GAN+SGM and GAN+MLP with 15 minutes window size and 75% overlap.

| Algorithm  | Precision | Recall | F1-measure | Accuracy |
|------------|-----------|--------|------------|----------|
| SGM        | 0.92      | 0.90   | 0.90       | 0.901    |
| GAN+SGM    | 0.98      | 0.97   | 0.97       | 0.970    |
| MLP        | 0.93      | 0.92   | 0.91       | 0.915    |
| GAN+MLP    | 0.97      | 0.96   | 0.95       | 0.958    |

when the augmented size is 15, 30, 60, and 120 minutes for all drivers. Then, we extract histogram features for all windows with 15 minutes window size and 75% overlap. Fig. 12 shows the results of these hybrid models. This figure shows that the results of all classifiers improve for 15 minutes augmentation size. Table II compares two hybrid models based on the standard machine learning measures. As one can see, GAN improves driving identification results in both hybrid models in accuracy, precision, recall, and F1-measure. Moreover, the highest accuracy is 97%, achieved by the hybrid model of GAN and SGM.

Now, we compare the proposed system with some driver identification systems stated in the related literature. For a fair comparison, the projects that use ECU or GPS data are left because of their differences in data modalities as they come from various standards. We focus on inertial sensors consuming the lowest computational burden, and in comparison with ECU sensors, their ubiquity is more comprehensive in terms of smartphone-embedded; also, GPS violates privacy.

Fig. 13 compares the proposed system with three baseline driver identification models based on inertial data, including Virojboonkiate et al. [17], Sánchez et al. [5], and Li et al. [26] to study how the accuracy diminishes as the number of drivers increases. We have implemented these baseline models in [1], according to their descriptions. The window size and overlap were 15 minutes and 75%. We compared the results of these methods on the dataset of [19] with ten drivers. As one can see, our system causes the most stability when the number of drivers grows. Really, the statistical pair of (average, standard deviation) of GAN+SGM is (99.17, 1.16) that is better than (75.96, 4.08) for [26], (44.57, 14.27) for [17], and (15.65, 4.95) for [5]. Thus, the accuracy of GAN+SGM is not sensitive to the number of drivers. Our results also are competitive with that of [28] that received to 99.99%, 99.7%, 99.6%, 99.5% accuracies for 5, 15, 35 and 50 drivers identification by ECU signal processing.

Fig. 13: Accuracy diminishing when the number of drivers grows up.

IV. CONCLUSION

In this paper, we proposed a hybrid model for driver identification, where Generative Adversarial Network (GAN) is used for data augmentation, and Stacked Generalization Method (SGM) is applied for the classification. GAN has been used for data augmentation in image processing context, while it is a novel for driving signals that process on the Discrete Wavelet Transform (DWT) of the accelerometer and gyroscope data. SGM is also an ensemble method that combines successful classifiers in an integrated system with the logistic regression aggregating strategy. We divided the training data and the augmented data into some overlapped windows with 15 minutes length and 75% overlap. Then the histograms of these windows were fed to SGM. The
accuracy, precision, recall, and F1-measure of the proposed hybrid model on ten drivers were 97%, 98%, 97%, and 97%, respectively. The experimental results showed that the sensors’ longitudinal axis is the most informative part. Our system’s hybrid model on ten drivers were 97%, 98%, 97%, and 97%, respectively. The experimental results showed that the sensors’

**REFERENCES**

[1] K. H. Park and H. K. Kim, “This car is mine!: Automobile theft countermeasure leveraging driver identification with generative adversarial networks,” arXiv preprint arXiv:1911.09870, 2019.

[2] J. R. López, L. C. González, J. Wahlström, M. M. y Gómez, L. Trujillo, and G. Ramírez-Alonso, “A genetic programming approach for driving score calculation in the context of intelligent transportation systems,” IEEE Sensors Journal, vol. 18, no. 17, pp. 7183–7192, 2018.

[3] S. Jafarnejad, G. Castignani, and T. Engel, “Towards a real-time driver identification mechanism based on driving sensing data,” in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017, pp. 1–7.

[4] M. A. Rahim, J. Liu, Z. Zhang, L. Zhu, X. Li, and S. Khan, “Who is driving? event-driven driver identification and impostor detection through support vector machine,” IEEE Sensors Journal, vol. 20, no. 12, pp. 6552–6559, 2020.

[5] S. H. Sánchez, R. F. Pozo, and L. A. H. Gómez, “Driver identification and verification from smartphone accelerometers using deep neural networks,” IEEE Transactions on Intelligent Transportation Systems, 2020.

[6] R. R. M. Putri, C.-H. Yang, C.-C. Chang, and D. Liang, “Smartwatch-based open-set driver identification by using GMM-based behavior modeling approach,” IEEE Sensors Journal, 2020.

[7] X. Zhang, Z. Zhou, and J. Rong, “A study of individual characteristics of driving behavior based on hidden markov model,” Sensors & Transducers, vol. 167, no. 3, p. 194, 2014.

[8] N. C. Fung, B. Wallace, A. D. Chan, R. Goubran, M. M. Porter, S. Marshall, and F. Knoefel, “Driver identification using vehicle acceleration and deceleration events from naturalistic driving of older drivers,” in 2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA). IEEE, 2017, pp. 33–38.

[9] M. Martínez, J. Echanobe, and I. del Campo, “Driver identification and impostor detection based on driving behavior signals,” in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016, pp. 372–378.

[10] F. Tahmarsi, Y. Wang, Y. Chen, and M. Gruteser, “Your phone tells us the truth: Driver identification using smartphone on one turn,” in Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, 2018, pp. 762–764.

[11] D. Hallac, A. Sharang, R. Stahlmann, A. Lamprecht, M. Huber, M. Roeher, J. Leskovec et al., “Driver identification using automobile sensor data from a single turn,” in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016, pp. 953–958.

[12] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “Autoaugment: Learning augmentation policies from data,” arXiv preprint arXiv:1805.09501, pp. 1–14, 2018.

[13] L. Moreira-Matias and H. Farah, “On developing a driver identification methodology using in-vehicle data recorders,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 9, pp. 2387–2396, 2017.

[14] H. R. Eftekhari and M. Ghaee, “Hybrid of discrete wavelet transform and adaptive neuro fuzzy inference system for overall driving behavior recognition,” Transportation Research Part F: Traffic Psychology and Behaviour, vol. 58, pp. 782–796, 2018.

[15] A. El Mekki, A. Bouhoute, and I. Berrada, “Improving driver identification for the next-generation of in-vehicle software systems,” IEEE Transactions on Vehicular Technology, vol. 68, no. 8, pp. 7406–7415, 2019.

[16] A. Chowdhury, T. Chakravarty, A. Ghose, T. Banerjee, and P. Bala-chait, “Journal of Advanced Transportation”, Journal of Advanced Transportation, vol. 5, no. 1, p. 9, 2018.

[17] A. Chowdhury, T. Chakravarty, A. Ghose, T. Banerjee, and P. Balachait, “Journal of Advanced Transportation,” Journal of Advanced Transportation, vol. 2019, 2019.

---

**Ruhallah Ahmadian** is currently a Ph.D. student with the Department of Computer Science, Amirkabir University of Technology. His research interests cover intelligent transportation systems, deep learning, generative adversarial networks, and optimization.

**Mehdi Ghaee** is an Associate Professor with the Department of Computer Science, Amirkabir University of Technology, Tehran, Iran. He has written more than 130 papers on national and international journals and conferences, two textbooks and two book chapters. Currently, he is director of NORC.

**Johan Wahlström** is an Assistant Professor with the Dept. of Computer Science, University of Exeter, UK. He was a Postdoc Researcher at Oxford University between January 2018 and June 2020, working on indoor navigation for emergency responders. Johan received his MSc degree in Engineering Physics and Ph.D. degree in Electrical Engineering from KTH Royal Institute of Technology, Stockholm, Sweden, in 2014 and 2017, respectively.