A detection model of aggressive driving behavior based on hybrid deep learning

Noor Walid Khalid, Wisam Dawood Abdullah
College of Computer Science and Mathematics, Tikrit University, Tikrit, Iraq

Article Info

Article history:

Received Jan 17, 2024
Revised Mar 4, 2024
Accepted Mar 21, 2024

Keywords:

Aggressive driving behavior
Convolution neural network
Deep learning
Dense
Feature selection

ABSTRACT

A major problem in today's transportation systems is driving behavior, since there are growing worries concerning ensuring the safety of motorists, passengers, and other road users. Deep learning algorithms can classify people based on their driving behaviors and identify driving trends from sensor data. This paper presents a novel model based on a driving behavior dataset gathered from cellphones for detecting and classifying aggressive driving. The model uses a hyper-deep learning model to create a prediction model that classifies drivers into three groups: normal, slow, and aggressive. The system starts with pre-processing methods normalization and standard scaler approaches to prepare the data. Two methodologies are used: directly entering the data into the deep model to classify driving behavior and selecting features using principal component analysis (PCA), singular value decomposition (SVD), and mutual information (MI). The hyper-convolutional neural network (CNN)-dense model is then used to train features to classify driver behavior. The experimental results show that the CNN-dense model with feature selection techniques SVD6 and MI6 achieves the best results with 100% accuracy rate for aggressive driver behavior detection, while the time for SVD6 is the shortest at 43 seconds.

Corresponding Author:

Noor Walid Khalid
College of Computer Science and Mathematics, Tikrit University
Tikrit, Iraq
Email: noorwalid1995@gmail.com

1. INTRODUCTION

Driving behaviors are the most common cause of traffic accidents and a large contribution to insurance claims [1]. There have been traffic accidents since Karl Benz invented the vehicle. The number of cars on the road increases in tandem with the economy and society, contributing to an increase in traffic accidents and congestion [2]. According to research, human factors are responsible for about 90% of roadway accidents [3]. Driving style can be described as a driver's habitual driving behavior that reflects their tendency to operate in particular ways regularly [4]. It also describes how a driver's style of driving affects both their own and other drivers' safety through driving [5]. Abnormal driving is defined as abnormal or unsafe behavior that deviates from the norms for a specific set of drivers [6]. There are other types of irregular driving, but the most relevant behaviors, like speeding, aggressive driving, and careless driving, are related to an increased chance of an accident [7]. Road rage is characterized by verbal abuse, shoving, hitting, threatening behavior, and maybe minor or major injuries [8]. It is described as a short-lived, intense emotional response to perceived provocation in a conflict situation involving two or more individuals on the road [9]. Speeding, tailgating, weaving in and out of traffic, and running red signals are all examples of aggressive driving [10]. According to a survey done by the american automobile association (AAA) foundation for traffic safety, aggressive driving behavior (ADB) was implicated in roughly 55.7% of fatal traffic accidents [11], and the frequency of road accidents and ADB are positively correlated [12]. ADB, as one of the leading causes of traffic issues, is influenced by both situational conditions like traffic congestion [13] and human ones like negative emotions [14]. Because of the progressively congested traffic system and the rapid pace of life, it is easier for drivers to display ADB, so proper recognition of ADB is critical. However, no single definition of ADB exists [15]. Interventions of technology in highway rage and aggressive driving are critical to achieving this goal [16]. Deep learning has seen fast development in the field of driving behavior identification in recent years [17]. It can help when a model is difficult to train due to a small sample size or when data collection is problematic in the target domain [18]. Deep learning has been used in various study domains due to its usual advantages. Figure 1 depicts the characteristics that influence driving behavior [19]. This paper aims to present a method for the detection of ADB in vehicles, focusing on developing a deep learning model by implementation a convolutional neural network (CNN) to the identification and classification of driving behaviors, with a focus on investigating how feature selection strategies affect model performance, this is something that previous studies did not give much attention to it. We will conduct a comparative analysis between the CNN model with feature selection and the model without feature selection,evaluate and quantify the impact of employing feature selection techniques on key performance metrics to discern the effectiveness of these methods. Demonstrate how the proposed deep learning model contributes to advancements in the field of driving behavior classification. Present new insights, improved methodologies, and potential applications that can significantly enhance the detection and understanding of ADB. Highlight advancements achieved and showcase the model's higher performance.

The remainder of this paper is structured as follows: section 2 provides an in-depth examination of driving behavior detection and deep learning applications. Section 3 describes the ADB detection mechanism, which is based on hyper-deep learning. Section 4 provides the comparison findings and a discussion of the implementation of the proposed deep ADB detection model with and without using feature selection. Section 5 summarizes the conclusions of this study.

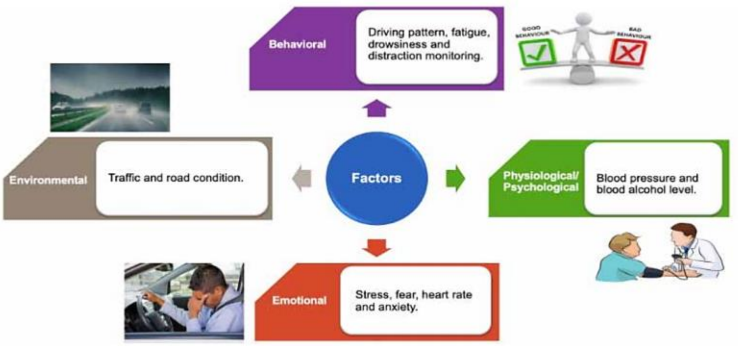

Figure 1. Factors influencing driving behavior [19]

2. LITERATURE SURVEY

This section includes a comprehensive review of literature ranging from representative works ranging from the oldest to the latest around this study. Several ways to detect driving behavior have been proposed over the last two decades. Moukafih *et al*. [20] proposed aggressive driver behavior classification model using long short-term memory (LSTM)-fully convolutional network (FCN) with real-world driving data from mobile phones. The UAH-drive set dataset is used to validate the technique. The method outperforms other deep learning and conventional machine learning models in terms of accuracy, with a 95.88% accuracy score for a 5-minute window duration. Matousek *et al*. [21] focused on developing a reliable method for identifying unusual driving behavior using neural networks. They compare LSTM networks and AutoEncoder replicator neural networks to an isolation forest. They show that a recurrent neural network (RNN) can reliably detect anomalies in driving behavior, with an accuracy rate of 93%, making it suitable for large-scale detection systems. Xing *et al*. [22] developed a RNN to address driver behavior profiling as an anomaly detection problem. The model, trained on data from typical drivers, produced significant regression error when predicting ADB, but low error when recognizing regular driving behavior. The model achieved an accuracy rate of 88% when classifying ADB, suggesting it could be a useful baseline for unsupervised driver profiling and contributing to a smart transportation ecology. Talebloo *et al*. [23] proposed a method to detect ADB using GPS sensors on smartphones. They classify drivers' driving behavior every three minutes using RNN algorithms, ignoring road conditions or driver's behavior. The algorithm, which uses 120 seconds of GPS data, has a 93% accuracy rate in identifying violent driving behavior, indicating that three minutes or more of driving is sufficient. Al-Hussein *et al*. [24] presented a method for profiling driver behavior using segment labeling and row labeling. A safety grade is assigned by row labeling to every second of driving data, while segment labeling grades temporal segments based on norms. The research uses three deep-learning-based algorithms: deep neural network (DNN), RNN, and CNN to classify recorded driving data. CNN was suggested for the system of identification, outperforming the other two techniques with 96.1% accuracy. The study suggests that this recognition system could increase road safety. The research aims to avoid overfitting and improve road safety.

Al-Hussein *et al.* [24] proposed an ADB recognition technique using collective learning. The majority class is grouped using a self-organizing map and linked with the minority class to create multiple class-balancing datasets. The classifiers are built using CNN, LSTM, and gated recurrent unit (GRU) techniques. The ensemble classifier is better suited for identifying ADBs in a tiny percentage of the dataset, while the classifier without ensemble learning is better for detecting more abundant ADBs. The LSTM and product rule-based ensemble classifier has the highest accuracy of 90.5% [25]. Escottá *et al*. [26] used inertial measurement unit (IMU) sensors on smartphones to identify driving events using linear acceleration and angular velocity signals. They evaluated deep-learning models using 1D and 2D CNNs, achieving high accuracy values of up to 82.40%. Cojocaru *et al.* [27] presented a deep learning-based driving behavior estimation system integrated into a ride-sharing application. Results that used the driving behavior dataset show better accuracy with two classes, with CNN-LSTM achieving the best results at 91.94%, and ConvLSTM outperforming classical LSTM networks [27]. Cojocaru and Popescu [28] showed a dataset collected utilizing an Android smartphone that exclusively utilizes sensor data from the smartphone. The dataset is classified into three categories: slow, normal, and aggressive, and it is accompanied by experiments aimed at offering insight into the data capacity. They proposed CNN, LSTM, and ConvLSTM models using three machine learning techniques. The results show that ConvLSTM achieved the highest accuracy of 79.5%. Abosaq *et al*. [29] suggested deep learning-based detection methods for anomalous driving behavior using a dataset with five categories. The proposed CNN-based model outperforms pre-trained models in performance metrics, achieving 89%, 93%, 93%, 94%, and 95% accuracy in classifying driver's unusual conduct.

3. METHODOLOGY

The methodology used in this study to combine feature reduction with rapid hyper-deep learning methods for precise classification of ADB is presented in this section of the paper. The methodical process employed to create a strong system that can reliably and precisely recognize ADB is described in this section. The model accurately classifies driving behaviors into three categories: slow, normal, and aggressive. It does this by utilizing feature selection and reduction approaches in conjunction with the capabilities of DNN. The model's ability to discriminate between different behavior categories with accuracy can support proactive efforts to improve traffic control and road safety. The next sections explain the procedures, evaluation strategies, and methodologies employed in this research project, going into detail about each stage of the process. The system's components, which include the driving behavior component, are shown in Figure 2.

3.1. Driving behavior dataset description

Our main objective is to present a thorough comprehension of the dataset used in our study in the section devoted to dataset gathering and description. We understand that the effectiveness of deep learning models depends critically on high-quality data. We shall provide comprehensive information in the ensuing subsections to achieve this goal. Important details like the data gathering process, the sources it came from, and the data cleaning and preprocessing steps will all be covered in our investigation. We hope that this thorough explanation will provide readers with a strong basis for comprehending the context of the dataset and its significance to our research. The dataset used in this study closely matches the goals of the research as well as the requirement for high-quality data to train hyper-deep model. Our study focuses on detecting and classifying driving behavior into three groups: normal, aggressive, and slow. Eight features make up the dataset that the application uses [27], [28]: i) three for the acceleration in meters per second squared on X, Y, and Z axes; ii) three (X, Y, Z) axes rotation in degrees per second (°/s); iii) label for classification (aggressive, normal, slow); and iv) date and time stamp. Only the accelerometer and gyroscope were utilized as the primary sensors, and the data was gathered in samples (two samples per second) after the gravitational acceleration was eliminated.

The dataset used in this study was sourced from Kaggle, a popular online platform for sharing dataset1. The data collection process involved meticulous recording using a Samsung Galaxy S10 smartphone and a Dacia Sendero 1.4 MPI vehicle. In terms of the choice of vehicle for data collection, a standard car with 75 horsepower was selected. The geospatial coverage of the dataset is focused on the city of Craiova, located in the Dolj region of Romania. This specific region was chosen as the data collection area to provide a localized perspective and account for any unique characteristics or dynamics present in that location. The dataset employed in the proposed model is described in terms of its characteristics. The summarized information is presented in Table 1, which provides an overview of the dataset's attributes, values, and other pertinent characteristics.

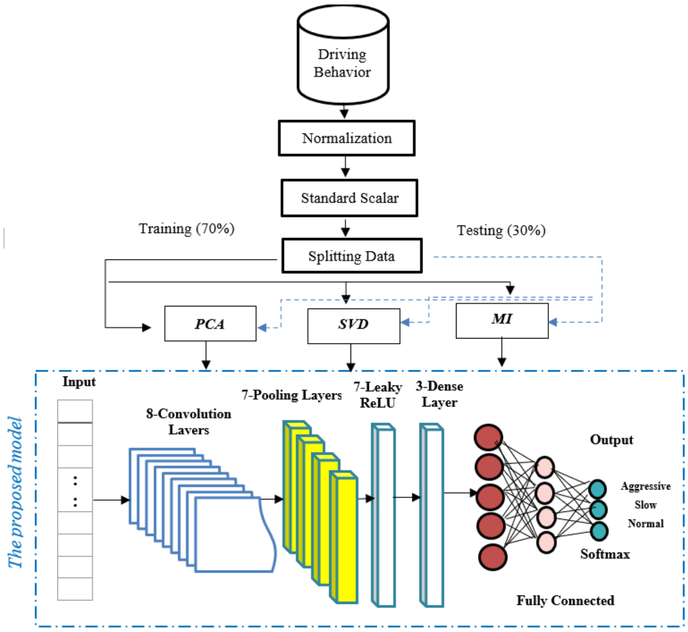

Figure 2. The proposed system architecture

Table 1. Characteristics and values of the dataset employed in the proposed model

Characteristic	Specification
Dataset name	Driving behavior
Number of samples	3644
Number of features	8
Missing data	No
Balanced dataset	Yes
Label	Yes

3.2. Data preprocessing

One of the most crucial phases of applications for data analysis is data preprocessing. Many inconsistencies, out-of-range numbers, missing values, noises, and/or excesses are among the numerous defects that are frequently present in raw data. Low-quality data will impede the learning and mining algorithms' ability to function well in the upcoming stages. Because of this, numerous preprocessing steps must be completed to improve the quality of raw data. Under this topic, some of the most popular and useful data preparation methods for use in data analysis applications are reviewed in terms of usage, popularity, and the algorithms that support them [30]. In this work, two commonly used techniques in data preprocessing were used. These techniques are normalization and standard scaler.

3.2.1. Normalize data

Normalize data: normalization, which involves scaling feature data to specific intervals such as [-1.0, 1.0] or [0.0, 1.0], is usually required when a dataset contains features with very different scales. If not, features with values on a much larger scale might make a smaller scaled but still significant feature less effective [31]. This will have a detrimental effect on the data mining model's accuracy performance. To equalize the size of the features, the normalizing technique is therefore done to them. The three most used techniques are decimal scale normalization, z-score normalization, and min-max normalizing [32]. Min-max normalization: The difference between the data's largest and lowest values is used to calculate the normalization. In (1) displays the values of the feature as min, max, and v, the values to be normalized, and the new range to be normalized is represented by $newmax$ and $newmin$ [33].

$$x_{new} = \frac{x - \min(x)}{\max(x) - \min(x)} (new_{max} \text{-} new_{min}) + new_{min} \tag{1}$$

Where x new represents normalized x. We implemented normalization techniques, notably min-max normalization, in our pre-processing due to its simplicity and effectiveness. When preserving the relationship between the original dataset is crucial, this method is especially helpful.

3.2.2. Standard scaler

Standard scaler, which implements Z-score normalization, standardizes characteristics by removing their mean from each value and dividing the outcome by the attribute's standard deviation s, producing a distribution with a mean of zero and a variance of one unit [34]. Let $\bar{x}$ be the mean of the x variable, and (2) transforms (scales) a value x_i into $\bar{x}_i$.

$$\bar{x}_i = \frac{x_i - \bar{x}}{s} \tag{2}$$

The translational word in this example is the attribute's sample mean, and the standard deviation serves as the scaling factor. This technique has the advantage of transforming both positive and negative-valued qualities into a relatively comparable distribution. However, when compared to an attribute without outliers, the final distribution of inliers is excessively narrow when outliers are present [35]. Standard scaler is used in this system to resize the value distribution so that the mean of the observed data is 0 and the standard deviation is 1.

3.3. Dataset splitting

Dataset splitting is a strategy that is widely regarded as essential for removing or reducing bias in training data in deep learning models. Data scientists and analysts always use this method to keep machine learning techniques from overfitting and underperforming on real test data [36]. Large datasets are typically divided into several well-defined subgroups by data scientists and analysts, who then use these subsets to train different parameters. The goal of this study is to determine which machine learning system parameters best fit the training data by considering the significant impact of splitting a dataset into multiple train sets and test sets [37]. In order to assess the predictive abilities of classification models, a clean dataset must be used for testing. As a result, the original dataset is divided into two subsets: the test dataset comprises 30% of the total observations, and the training dataset comprises 70% of the total observations in the original dataset. The test dataset is kept clean so that model detection may be made on it, while the training dataset is utilized to train the model and fine-tune parameters.

Finding a balance between a suitably large training set and an equally sizable testing set was the major criterion that guided our dataset splitting which offering a solid assessment of the generality of the model. By setting aside 70% of the dataset for training, allowing the model to become familiar with and adjust to the underlying patterns in the data. In addition, setting aside 30% for testing guarantees a sizable collection of unknown cases for assessing the model's effectiveness, achieving a balance between model learning and assessment. The model is less likely to overfit since it has enough data to comprehend underlying patterns without learning noise, thanks to the bigger part (70%) that is devoted to training. We aim to improve the transparency and credibility of our results in the field of aggressive driver behavior identification by using this method.

3.4. Feature relevance assessment methods

A preprocessing technique that determines essential attributes of a problem is feature selection. Reducing the number of features, which means the number of columns in a dataset is the primary method used to do it. The model's accuracy rate and inference quality increase as the number of features is decreased without compromising the quality of the dataset, while learning time and available space are decreased. To give these advantages, many feature selection algorithms are available. Three methods were employed in the suggested model: principal component analysis (PCA), singular value decomposition (SVD), and mutual information (MI). In this section, more details about these techniques will be explained.

3.4.1. Using principal component analysis to select features (1st technique)

Using PCA to select features (1st technique): The first technique used in this system is PCA, PCA is a transformation approach that reduces the size of a dataset by transforming it into fewer associated variables [38]. PCA is a decomposition of a column-mean-centered data matrix X of size N×K, where N and K are the number of samples and features, respectively.

$$x = TP^T + E \tag{3}$$

T is a scoring matrix of size N×A connected to the matrix X projections into an A-dimensional space, P is a loading matrix of size K×A related to the feature projections into an A-dimensional space (with $P^T P$=I), and E is a residual matrix of size N×K [39].

3.4.2. Using singular value decomposition to select features (2nd technique)

The second technique utilized in the proposed system to select the best features that make the accuracy of detection and classification almost identical is SVD, as PCA but most specifically, the initial A principal components and the SVD of X are used to identify the A-dimensional space. When we denote X=USV^T as the SVD of X and $\widehat{U}$, $\hat{S}$, and $\hat{V}$ as the matrices containing the first A columns of U, S, and V, respectively, we get:

$$T = \widehat{U} \times \hat{S} \tag{4}$$

$$P = \hat{V} \tag{5}$$

And X=TP^T is named the reconstructed data matrix [40].

3.4.3. Using mutual information to select features (3rd technique)

The third technique used in the proposed model to increase the accuracy and decrease the time of execution is MI. Studies on MI dating from early to the 1990s show that it is one of the most popular feature selection techniques [41]. By calculating how much data about one random feature can be obtained from the other, MI quantifies the mutually dependent relationship between two random features. It is therefore associated with the entropy of a random feature, which is established by the quantity of information included in the feature. The MI between two discrete random variables X and Y is defined to be as (6) [42].

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log_2 (\frac{P(x,y)}{p(x)p(y)}) \tag{6}$$

Three separate feature selection techniques were carefully selected in the study to handle the particular difficulties involved in identifying ADB. Each of these strategies has unique benefits that complement the research objectives and increase the stability and efficacy of the suggested multi-stage system. It was decided to combine PCA, SVD, and MI in order to the harness advantages of each technique. The driving behavior dataset has high dimensionality, and because PCA effectively lowers dimensionality and preserves important information, it is a good fit for our study since our objective is to discover driving behavior's influential features. A different viewpoint on the latent structures in the dataset is offered by SVD, which enhances PCA. Capturing subtle correlations in driving behavior features was the motivating force behind its usage MI was selected in order to evaluate the information gained related to several characteristics in relation to ADB. The intricacy of driving behavior datasets is in accordance with its capacity to manage non-linear interactions.

3.5. Convolution neural network to classify data

CNN, also known as ConvNet, is a kind of artificial neural network (ANN) with remarkable generalization capabilities and a deep feed-forward design [43]. It can learn highly abstracted features of things, especially spatial data, and recognize them more effectively than other networks with FC layers [44]–[46]. A deep CNN model consists of a limited number of processing layers that can be trained at different levels of abstraction to learn different features of input data (like images) [47]. Higher abstraction is achieved by the deeper layers in learning and extracting low-level data, while lower abstraction is achieved by the initiatory levels [48]. Figure 3 depicts the conceptual form of the proposed CNN-dense, with different sorts of layers discussed in the following section.

- Convolution layer: the convolutional layer is the most crucial part of any CNN architecture. To create an output feature map, it consists of a set of convolutional kernels, sometimes referred to as filters, convolved with the input image (N-dimensional metrics) [49], [50].
- Pooling layer: layers sub-sample feature maps produced after convolution operations, preserving dominant features in each pool step. Pooling operations specify the pooled region size and stride, like convolution. Different techniques like max pooling, min pooling, average pooling, gated pooling, and tree pooling are used in different layers, with max pooling being the most popular and commonly used technique [51]–[53].
- Leaky ReLU: this activation function, in contrast to ReLU, downscales the negative inputs rather than totally ignoring them. The Dying ReLU problem is resolved by using leaky ReLU. leaky ReLU is represented mathematically as (7) [54]:

$$F(x)_{Leaky\ ReLU} = \begin{cases} x \ if \ x > 0 \\ mx \ x \leq 0 \end{cases} \tag{7}$$

Where m is a constant, also known as the leak factor, and is often set to a low number (e.g., 0.001).

- Dense*:* this layer of a standard DNN is what it is called. It is the most often used and common layer. The following process is carried out on the input by the dense layer, which then returns the outcome. The formulation of this layer is (8) [55]:

$$Output \ = \ Activation\ (dot\ (input, kernel)\ +\ bias) \tag{8}$$

- Flatten: the output of the pooling layer will be a matrix, which the neural network cannot receive. The n×n matrix from the pooling layer is converted into n^2×1 matrix by the flattening layer so that it may be fed into the neural network [56].
- Fully connected layers*:* in a CNN model, one or more fully connected layers are often included just before the classification output. Similar to neural network layer topologies, neurons between neighboring layers are fully connected, and a completely connected layer consists of a fixed number of disconnected neurons [57], [58].

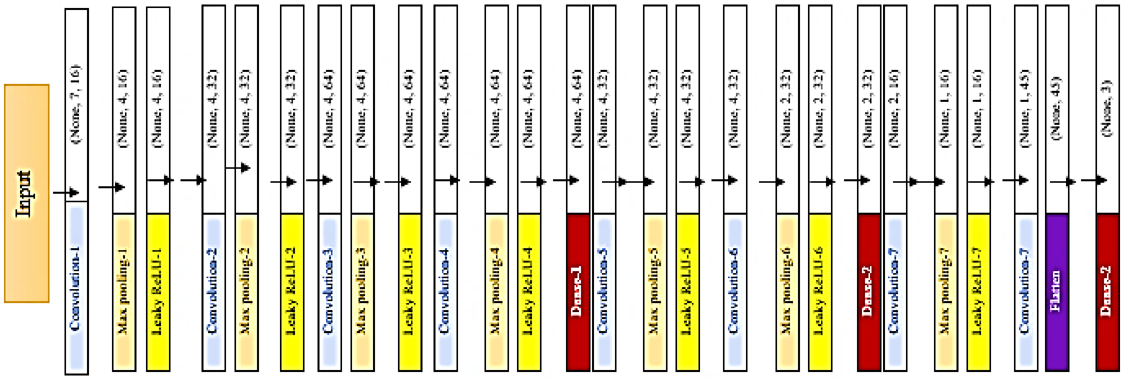

Figure 3. Architecture of the proposed CNN-Dense model

3.5.1. The proposed convolutional neural network-dense model for driver behavior detection and classification

The proposed CNN-Dense Model for driver behavior detection and classification: The proposed CNN-Dense model for ADB is explained in this section. The proposed CNN model is utilized to classify data immediately after the dataset is loaded, processed, and split in this technique. The suggested CNN-dense model has 26 layers, which are as follows: i) CNN with 8 layers, ii) leaky ReLU with 7 layers, iii) Max Pooling with 7 layers, iv) 1 layer should be flattened, and v) dense is 3 layers. Table 2 goes into much detail about these layers.

Table 2. The proposed hyper CNN-dense layers

NO.	Layer type	Filters	Size/Stride	Activation function	#Param
1	Convolutional	16	3/1	–	64
3	Max Pooling	–	2/2	–	0
3	Leaky ReLU	–	–	–	0
4	Convolutional	32	3/1	–	1568
5	Max Pooling	–	2/1	–	0
6	Leaky ReLU	–	–	–	0
7	Convolutional	64	3/1	–	6208
8	Max Pooling	–	2/1	–	0
9	Leaky ReLU	–	–	–	0
10	Convolutional	64	3/1	–	12352
11	Max Pooling	–	2/1	–	0
12	Leaky ReLU	–	–	–	0
13	Dense	64	–	Linear	4160
14	Convolutional	32	3/1	–	6176
15	Max Pooling	–	2/1	–	0
16	Leaky ReLU	–	–	–	0
17	Convolutional	32	3/1	–	3104
18	Max Pooling	–	2/2	–	0
19	Leaky ReLU	–	–	–	0
20	Dense	32	–	Linear	1056
21	Convolutional	16	3/1	–	1552
22	Max Pooling	–	2/2	–	0
23	Leaky ReLU	–	–	–	0
24	Convolutional	45	3/1	–	2205
25	Flatten	–	–	–	0
26	Dense	32	–	Softmax	138

4. RESULT AND DISCUSION

In this section, we present our research findings and provide a thorough analysis and interpretation of them in the context of our study objectives. We have divided this section into several subsections to make sure the presentation is well-organized. Two methodologies were used in the proposed model as follows.

4.1. Classify data using hyper CNN-dense without feature selection "1st methodology"

In this methodology, the data set is first processed using two pre-processing techniques, then the data is separated into two groups, the first is used to train the proposed model and the other is used for testing. The data is entered as is to the classification stage and the results of this stage using evaluation metrics [59], [60] are shown in Tables 3 and Figure 4.

Table 3. The results of proposed CNN-dense without feature selection

Technique	Accuracy	Precision	Recall	f-measure	Time in sec.
CNN-Dense	95.2%	95%	94.7%	94.8%	41

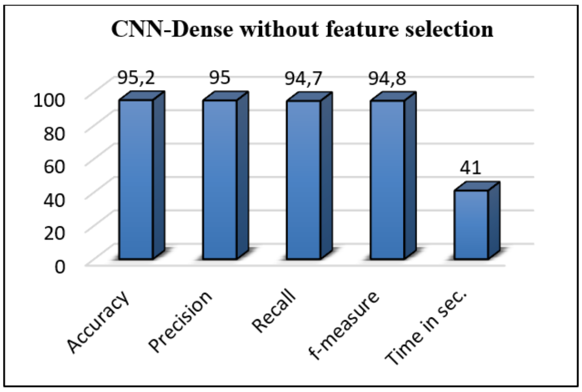

Figure 4. Chart of results of proposed CNN-dense without feature selection

4.2. Classify data using hyper CNN-dense using feature selection "2nd methodology"

A feature selection is merely choosing or eliminating specific features without altering them in any manner. Dimensionality reduction is the process of reducing the dimensionality of features. The set of features produced by feature selection, on the other hand, must be a subset of the original set of features. The set produced by dimensionality reduction does not have to be (for example, PCA decreases dimensionality by generating new synthetic features by linearly mixing the existing features and removing the less significant ones). In this sense, feature selection is a subset of dimensionality reduction. Feature selection and reduction approaches were employed in this study to improve the efficiency of our suggested hyper CNN-Dense model. This section digs into how various strategies affect model performance and computational complexity. The emphasis is on identifying the most useful traits and how they contribute to improved prediction accuracy. Table 4 and Figure 5 display the results of three feature selection strategies (PCA, SVD, and MI) combined with the proposed CNN-dense model.

Table 4. The results of proposed CNN-Dense with PCA, SVD, and MI feature selection

Technique	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Time in second
PCA3	75.4	78.5	78.3	78.3	30
PCA4	96.8	98.4	98.4	98.4	54
PCA5	85.7	82.4	82.4	82.3	14
PCA6	98.7	98.9	98.9	98.9	24
SVD3	73	75.1	75	75	36
SVD4	97.6	97.6	97.6	97.6	48
SVD5	99.9	99.9	99.9	99.9	40
SVD6	100	100	100	100	43
MI3	70.5	73.7	72.9	72.8	44
MI4	91.8	94.5	94.5	94.5	50
MI5	99	99.3	99.3	99.3	59
MI6	100	100	100	100	51

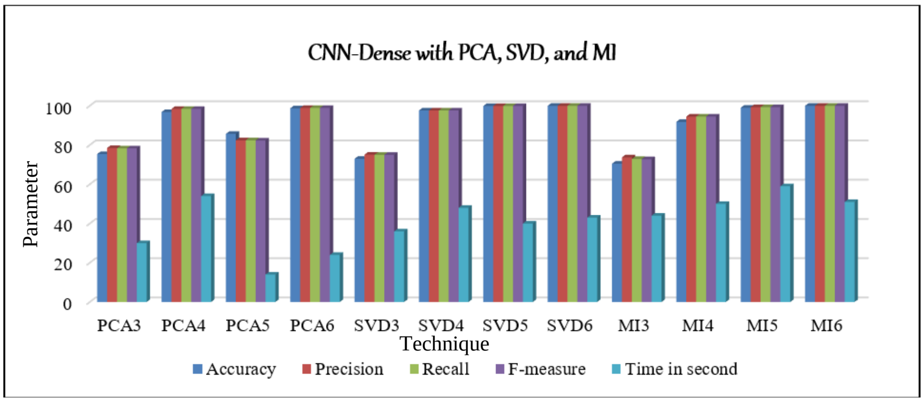

Figure 5. Chart of suggested CNN-dense results with feature selection

The suggested model with SVD6 and MI6 produced the best results, even when utilizing feature selection techniques with a 100% accuracy rate for aggressive driver behavior detection, while the time for SVD6 was the shortest, at 43 seconds. Feature selection is used in the deep learning process to improve accuracy. It also improves the detection capacity of the algorithms by identifying the most important variables and removing the redundant and irrelevant ones. This is why feature selection is so crucial. The following are three major advantages of feature selection:

– Reduces over-fitting: less duplicated data implies fewer opportunities to make conclusions based on noise.
– Improves accuracy: less misleading data implies more accurate modeling.
– Shortens training time: less data implies faster algorithms.

This study included three distinct feature selection strategies that were carefully chosen to address the unique challenges associated with classifying ADB. The distinct advantages of each of these approaches enhance the goals of the research while strengthening the stability and effectiveness of the proposed multi-stage system. The decision was made to use PCA, SVD, and MI to fully utilize the benefits of each method and then compare the results and determine the best. The Driving Behavior dataset is high dimensional, and since our goal is to identify the influential aspects of driving behavior, PCA successfully lowers dimensionality while preserving relevant information, making it a strong fit for our study. SVD improves PCA by providing an alternative perspective on the latent structures in the dataset. The driving force behind its use was the ability to identify tiny correlations in features associated with driving behavior. MI was chosen to assess the knowledge acquired on many traits associated with ADB. Driving behavior datasets are complex because of their ability to handle non-linear interactions. The drawbacks of these approaches include limited interpretability of the major component in terms of original features. For SVD, the dataset was sensitive to noise, and MI required a lot of computation, particularly for big feature sets.

4.3. Results comparison

When comparing the results obtained from the proposed hyper CNN-dense system with the results of previous studies that worked on the same dataset in Table 5 and Figure 6, we notice the superiority of the proposed model in all cases, even using the first methodology without feature extraction the accuracy result was 95.2%. In other cases, when feature extraction techniques were used the results obtained for accuracy were 100% with SVD6 and MI6 as the best accuracy, and with other techniques the accuracy also reached 99.9% and 99% as well, and the rest of the results were also good compared to the results of previous studies [27], [28] that gave detection accuracy of 91.94% and 79.5% respectively when using the same Driving behavior dataset. In these two studies they didn't use feature selection techniques cause of this our detection accuracy was better by using three of fearure selection techniques (PCA, SVD, and MI). In addition, time of execution for our proposed system was few causes of these used appraoch.

Table 5. Comparison results on driving behavior dataset

Reference	Accuracy (%)
[27]	91.94
[28]	79.5
Our proposed CNN-dense	100

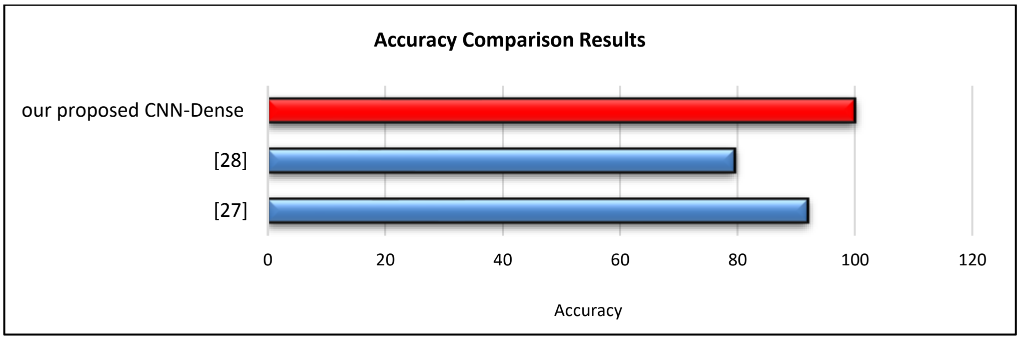

Figure 6. Comparison results with related works which used the same dataset

5. CONCLUSION

The accurate detection of ADB is the foundation for early and effective warning or assistance to the driver, which is critical for increasing driving safety. In this study, an ADB detection model based on hyper-deep learning CNN-dense is built using the driving behavior dataset; a proposed classify model is built; feature selection techniques are used; and the model is trained and tested using the driving behavior dataset obtained in a driving environment that is realistic. Results indicate that the proposed deep learning model achieves greater accuracy, prediction, recall, and F1-measure of 100% with SVD6 in 43 seconds and MI6 in 51 seconds. In contrast, the proposed model designed without feature selection achieved 95.2% accuracy in 41 seconds, where these results were the worest results for the proposed system. This comparison result indicates that the suggested model with feature selection is better suited for accurately detecting ADB, even with a limited part of the dataset. In terms of future work in this field, we should note that the dataset can be enhanced with data that can be measured to identify emotional, environmental, and psychological components rather than just behavioral factors. The proposed architecture enables its adaptation to diverse datasets and scenarios, making it a valuable asset for addressing various challenges in transportation, safety, and urban planning. Future applications can build on this research's foundation to further many aspects of intelligent systems and deepen our understanding of how people behave in dynamic contexts, such as use in expand the model's use beyond aggression analysis of driving behavior. Make use of the architecture to categorize and comprehend different driving behaviors, such as following traffic laws, being defensive, or driving while distracted. The capacity of the model to identify subtle driving patterns can help improve the way self-driving cars make decisions in intricate traffic situations.

REFERENCES

[1] S. Arumugam and R. Bhargavi, "A survey on driving behavior analysis in usage based insurance using big data," *Journal of Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0249-5.

[2] J. Hu, X. Zhang, and S. Maybank, "Abnormal driving detection with normalized driving behavior data: a deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6943–6951, Jul. 2020, doi: 10.1109/TVT.2020.2993247.

[3] C. Zhang, R. Li, W. Kim, D. Yoon, and P. Patras, "Driver behavior recognition via interwoven deep convolutional neural nets with multi-stream inputs," *IEEE Access*, vol. 8, pp. 191138–191151, 2020, doi: 10.1109/ACCESS.2020.3032344.

[4] E. Khosravi, A. M. A. Hemmatyar, M. J. Siavoshani, and B. Moshiri, "Safe deep driving behavior detection (S3D)," *IEEE Access*, vol. 10, pp. 113827–113838, 2022, doi: 10.1109/ACCESS.2022.3217644.

[5] M. Malik and R. Nandal, "A framework on driving behavior and pattern using on-board diagnostics (OBD-II) tool," *Materials Today: Proceedings*, vol. 80, pp. 3762–3768, 2023, doi: 10.1016/j.matpr.2021.07.376.

[6] C. Katrakazas, E. Michelaraki, M. Sekadakis, and G. Yannis, "A descriptive analysis of the effect of the COVID-19 pandemic on driving behavior and road safety," *Transportation Research Interdisciplinary Perspectives*, vol. 7, Sep. 2020, doi: 10.1016/j.trip.2020.100186.

[7] K. Wang, Q. Xue, Y. Xing, and C. Li, "Improve aggressive driver recognition using collision surrogate measurement and imbalanced class boosting," *International Journal of Environmental Research and Public Health*, vol. 17, no. 7, Mar. 2020, doi: 10.3390/ijerph17072375.

[8] Y. Ma, Z. Xie, S. Chen, F. Qiao, and Z. Li, "Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals," *Transportation Research Part C: Emerging Technologies*, vol. 146, Jan. 2023, doi: 10.1016/j.trc.2022.103983.

[9] Y. Zhang, Y. He, and L. Zhang, "Recognition method of abnormal driving behavior using the bidirectional gated recurrent unit and convolutional neural network," *Physica A: Statistical Mechanics and its Applications*, vol. 609, 2023, doi: 10.1016/j.physa.2022.128317.

[10] H. Zhu, R. Xiao, J. Zhang, J. Liu, C. Li, and L. Yang, "A driving behavior risk classification framework via the unbalanced time series samples," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2022.3145359.

[11] P. Wawage and Y. Deshpande, “Smartphone sensor dataset for driver behavior analysis,” *Data in Brief*, vol. 41, 2022, doi: 10.1016/j.dib.2022.107992.
[12] F. Guo, “Statistical methods for naturalistic driving studies,” *Annual Review of Statistics and Its Application*, vol. 6, no. 1, pp. 309–328, Mar. 2019, doi: 10.1146/annurev-statistics-030718-105153.
[13] M. Zahid, Y. Chen, S. Khan, A. Jamal, M. Ijaz, and T. Ahmed, “Predicting risky and aggressive driving behavior among taxi drivers: Do spatio-temporal attributes matter?,” *International Journal of Environmental Research and Public Health*, vol. 17, no. 11, Jun. 2020, doi: 10.3390/ijerph17113937.
[14] M. A. Khodairy and G. Abosamra, “Driving behavior classification based on oversampled signals of smartphone embedded sensors using an optimized stacked-LSTM neural networks,” *IEEE Access*, vol. 9, pp. 4957–4972, 2021, doi: 10.1109/ACCESS.2020.3048915.
[15] J. Hu, L. Xu, X. He, and W. Meng, “Abnormal driving detection based on normalized driving behavior,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 6645–6652, Aug. 2017, doi: 10.1109/TVT.2017.2660497.
[16] S. B. Brahim, H. Ghazzai, H. Besbes, and Y. Massoud, “A machine learning smartphone-based sensing for driver behavior classification,” in *IEEE International Symposium on Circuits and Systems*, May 2022, pp. 610–614, doi: 10.1109/ISCAS48785.2022.9937801.
[17] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, “Driver behavior detection and classification using deep convolutional neural networks,” *Expert Systems with Applications*, vol. 149, Jul. 2020, doi: 10.1016/j.eswa.2020.113240.
[18] P. Ping, C. Huang, W. Ding, Y. Liu, M. Chiyomi, and T. Kazuya, “Distracted driving detection based on the fusion of deep learning and causal reasoning,” *Information Fusion*, vol. 89, pp. 121–142, Jan. 2023, doi: 10.1016/j.inffus.2022.08.009.
[19] S. Arumugam and R. Bhargavi, “Road rage and aggressive driving behaviour detection in usage-based insurance using machine learning,” *International Journal of Software Innovation*, vol. 11, no. 1, pp. 1–29, Mar. 2023, doi: 10.4018/IJSI.319314.
[20] Y. Moukafih, H. Hafidi, and M. Ghogho, “Aggressive driving detection using deep learning-based time series classification,” in *IEEE International Symposium on INnovations in Intelligent SysTems and Applications, INISTA 2019*, pp. 1–5, Jul. 2019, doi: 10.1109/INISTA.2019.8778416.
[21] M. Matousek, M. El-Zohairy, A. Al-Momani, F. Kargl, and C. Bosch, “Detecting anomalous driving behavior using neural networks,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 2229–2235, Jun. 2019, doi: 10.1109/IVS.2019.8814246.
[22] Y. Xing, C. Lv, and D. Cao, “Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1341–1352, Feb. 2020, doi: 10.1109/TVT.2019.2960110.
[23] F. Talebloo, E. A. Mohammed, and B. H. Far, “Dynamic and systematic survey of deep learning approaches for driving behavior analysis,” *NSERC Discovery Grant and Alberta Ma jor Innovation Fund (MIF),* 2021, doi: 10.48550/arXiv.2109.08996.
[24] W. A. Al-Hussein, L. Y. Por, M. L. M. Kiah, and B. B. Zaidan, “Driver behavior profiling and recognition using deep-learning methods: in accordance with traffic regulations and experts guidelines,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 3, Jan. 2022, doi: 10.3390/ijerph19031470.
[25] H. Wang *et al.*, “A recognition method of aggressive driving behavior based on ensemble learning,” *Sensors*, vol. 22, no. 2, Jan. 2022, doi: 10.3390/s22020644.
[26] Á. T. Escottá, W. Beccaro, and M. A. Ramírez, “Evaluation of 1D and 2D deep convolutional neural networks for driving event recognition,” *Sensors*, vol. 22, no. 11, Jun. 2022, doi: 10.3390/s22114226.
[27] I. Cojocaru, P. Ș. Popescu, and M. C. Mihăescu, “Driver behaviour analysis based on deep learning algorithms,” in *RoCHI - International Conference on Human-Computer Interaction*, 2022, pp. 108–114, doi: 10.37789/rochi.2022.1.1.18.
[28] I. Cojocaru and P. Ș. Popescu, “Building a driving behaviour dataset,” in *RoCHI - International Conference on Human-Computer Interaction*, 2022, pp. 101–107, doi: 10.37789/rochi.2022.1.1.17.
[29] H. A. Abosaq *et al.*, “Unusual driver behavior detection in videos using deep learning models,” *Sensors*, vol. 23, no. 1, Dec. 2023, doi: 10.3390/s23010311.
[30] S. R. -Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, “A survey on data preprocessing for data stream mining: Current status and future directions,” *Neurocomputing*, vol. 239, pp. 39–57, May 2017, doi: 10.1016/j.neucom.2017.01.078.
[31] D. Singh and B. Singh, “Investigating the impact of data normalization on classification performance,” *Applied Soft Computing*, vol. 97, Dec. 2020, doi: 10.1016/j.asoc.2019.105524.
[32] Y. Chen *et al.*, “A deep learning model for the normalization of institution names by multisource literature feature fusion: algorithm development study,” *JMIR Formative Research*, vol. 7, Aug. 2023, doi: 10.2196/47434.
[33] A. Ali and N. Senan, “The effect of normalization in violence video classification performance,” *IOP Conference Series: Materials Science and Engineering*, vol. 226, no. 1, Aug. 2017, doi: 10.1088/1757-899X/226/1/012082.
[34] L. B. V. D. Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” *Applied Soft Computing*, vol. 133, Jan. 2023, doi: 10.1016/j.asoc.2022.109924.
[35] R. Dzierżak, “Comparison of the influence of standardization and normalization of data on the effectiveness of spongy tissue texture classification,” *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Srodowiska*, vol. 9, no. 3, pp. 66–69, Sep. 2019, doi: 10.35784/IAPGOS.62.
[36] Y. Xu and R. Goodacre, “On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning,” *Journal of Analysis and Testing*, vol. 2, no. 3, pp. 249–262, Jul. 2018, doi: 10.1007/s41664-018-0068-2.
[37] C. Yücelbaş and Ş. Yücelbaş, “Enhanced cross-validation methods leveraging clustering techniques,” *Traitement du Signal*, vol. 40, no. 6, pp. 2649–2660, Dec. 2023, doi: 10.18280/ts.400626.
[38] P. Rani, R. Kumar, A. Jain, R. Lamba, R. K. Sachdeva, and T. Choudhury, “PCA-DNN: a novel deep neural network oriented system for breast cancer classification,” *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 9, no. 1, Oct. 2023, doi: 10.4108/eetpht.9.3533.
[39] A. Malhi and R. X. Gao, “PCA-based feature selection scheme for machine defect classification,” *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 6, pp. 1517–1525, Dec. 2004, doi: 10.1109/TIM.2004.834070.
[40] X. Zhao and B. Ye, “Feature frequency extraction algorithm based on the singular value decomposition with changed matrix size and its application in fault diagnosis,” *Journal of Sound and Vibration*, vol. 526, May 2022, doi: 10.1016/j.jsv.2022.116848.
[41] M. A. Hossain and M. S. Islam, “A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection,” *Scientific Reports*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-48230-1.
[42] N. Barraza, S. Moro, M. Ferreyra, and A. D. L. Peña, “Mutual information and sensitivity analysis for feature selection in customer targeting: A comparative study,” *Journal of Information Science*, vol. 45, no. 1, pp. 53–67, Feb. 2019, doi: 10.1177/0165551518770967.

[43] T. T. Khoei, H. O. Slimane, and N. Kaabouch, "Deep learning: systematic review, models, challenges, and research directions," *Neural Computing and Applications*, vol. 35, no. 31, pp. 23103–23124, Nov. 2023, doi: 10.1007/s00521-023-08957-4.

[44] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 2, pp. 757–774, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.014.

[45] L. Alzubaidi *et al.*, "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," *Journal of Big Data*, vol. 10, no. 1, Apr. 2023, doi: 10.1186/s40537-023-00727-2.

[46] M. M. Taye, "Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, Mar. 2023, doi: 10.3390/computation11030052.

[47] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54–70, 2023, doi: 10.1016/j.cogr.2023.04.001.

[48] J. Dong, M. Zhao, Y. Liu, Y. Su, and X. Zeng, "Deep learning in retrosynthesis planning: Datasets, models and tools," *Briefings in Bioinformatics*, vol. 23, no. 1, Jan. 2022, doi: 10.1093/bib/bbab391.

[49] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, 2020, doi: 10.1007/s13748-019-00203-0.

[50] S. F. Ahmed *et al.*, "Deep learning modelling techniques: current progress, applications, advantages, and challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13521–13617, Nov. 2023, doi: 10.1007/s10462-023-10466-8.

[51] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[52] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[53] A. Mathew, P. Amudha, and S. Sivakumari, "Deep learning techniques: an overview," *Advances in Intelligent Systems and Computing*, vol. 1141, pp. 599–608, 2021, doi: 10.1007/978-981-15-3383-9_54.

[54] Y. Bai, "RELU-function and derived function review," *SHS Web of Conferences*, vol. 144, 2022, doi: 10.1051/shsconf/202214402006.

[55] V. L. H. Josephine, A. P. Nirmala, and V. L. Alluri, "Impact of hidden dense layers in convolutional neural network to enhance performance of classification model," *IOP Conference Series: Materials Science and Engineering*, vol. 1131, no. 1, Apr. 2021, doi: 10.1088/1757-899x/1131/1/012007.

[56] P. Chakraborty and C. Tharini, "Pneumonia and eye disease detection using convolutional neural networks," *Engineering, Technology and Applied Science Research*, vol. 10, no. 3, pp. 5769–5774, Jun. 2020, doi: 10.48084/etasr.3503.

[57] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.

[58] S. A. Suha and T. F. Sanam, "A deep convolutional neural network-based approach for detecting burn severity from skin burn images," *Machine Learning with Applications*, vol. 9, Sep. 2022, doi: 10.1016/j.mlwa.2022.100371.

[59] Ž. Vujović, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.

[60] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "multiclass confusion matrix reduction method and its application on net promoter score classification problem," *Technologies*, vol. 9, no. 4, Nov. 2021, doi: 10.3390/technologies9040081.

BIOGRAPHIES OF AUTHORS

Noor Walid Khalid is a candidate in the program of Master in Computer Science, Tikrit University. She received her B.Sc. degree in Computer Science from Tikrit University, in 2018. She is currently working as an employee in the laboratories of the College of Computer Science at Tikrit University. She can be contacted at email: noorwalid1995@gmail.com.

Wisam Dawood Abdullah is an associate professor and a faculty member at Tikrit University. He received his B.Sc. Degree in Computer Science from Tikrit University, and his M.S. degree in Information Technology (with a concentration in telecommunications and networks) from the University Utara Malaysia (UUM). He received an expert certification from Cisco Networking Academy CCNP, CCNA, CCNA security, IoT, entrepreneurship, grid, voice, wireless cloud, Linux, CCNA cybersecurity, and IT. In addition, he is a NetAcad administrator at Cisco Networking Academy. Recently, he is selected as AWS Community Builder at Amazon. His research interest includes protocol engineering, network analysis, cybersecurity, cloud computing, network traffic analysis, data mining, future internet, internet of things, AI, and ML. He can be contacted at email: wisamdawood@tu.edu.iq.