IEEE Access and Machine Learning,Volume:10,Issue Date:12.April.2022

Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms

FAWAZ KHALED ALARFAJ[1], IQRA MALIK[2], HIKMAT ULLAH KHAN[3], NAIF ALMUSALLAM[1], MUHAMMAD RAMZAN[2], AND MUZAMIL AHMED[3]

[1]Department of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11564, Saudi Arabia
[2]Department of Computer Science and Information Technology, University of Sargodha, Sargodha 40100, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Wah Campus, Wah Cantt 47040, Pakistan

Corresponding author: Hikmat Ullah Khan (hikmat.ullah@ciitwah.edu.pk)

This work was supported by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University through the Research Group under Grant RG-21-51-01.

ABSTRACT People can use credit cards for online transactions as it provides an efficient and easy-to-use facility. With the increase in usage of credit cards, the capacity of credit card misuse has also enhanced. Credit card frauds cause significant financial losses for both credit card holders and financial companies. In this research study, the main aim is to detect such frauds, including the accessibility of public data, high-class imbalance data, the changes in fraud nature, and high rates of false alarm. The relevant literature presents many machines learning based approaches for credit card detection, such as Extreme Learning Method, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression and XG Boost. However, due to low accuracy, there is still a need to apply state of the art deep learning algorithms to reduce fraud losses. The main focus has been to apply the recent development of deep learning algorithms for this purpose. Comparative analysis of both machine learning and deep learning algorithms was performed to find efficient outcomes. The detailed empirical analysis is carried out using the European card benchmark dataset for fraud detection. A machine learning algorithm was first applied to the dataset, which improved the accuracy of detection of the frauds to some extent. Later, three architectures based on a convolutional neural network are applied to improve fraud detection performance. Further addition of layers further increased the accuracy of detection. A comprehensive empirical analysis has been carried out by applying variations in the number of hidden layers, epochs and applying the latest models. The evaluation of research work shows the improved results achieved, such as accuracy, f1-score, precision and AUC Curves having optimized values of 99.9%,85.71%,93%, and 98%, respectively. The proposed model outperforms the state-of-the-art machine learning and deep learning algorithms for credit card detection problems. In addition, we have performed experiments by balancing the data and applying deep learning algorithms to minimize the false negative rate. The proposed approaches can be implemented effectively for the real-world detection of credit card fraud.

INDEX TERMS Fraud detection, deep learning, machine learning, online fraud, credit card frauds, transaction data analysis.

I. INTRODUCTION

Credit card fraud (CCF) is a type of identity theft in which someone other than the owner makes an unlawful transaction using a credit card or account details. A credit card that has been stolen, lost, or counterfeited might result in fraud. Card-not-present fraud, or the use of your credit card number in e-commerce transactions has also become increasingly common as a result of the increase in online shopping. Increased fraud, such as CCF, has resulted from the expansion of e-banking and several online payment environments, resulting in annual losses of billions of dollars. In this era of digital payments, CCF detection has become one of the most important goals. As a business owner, it cannot be disputed

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.

that the future is heading towards a cashless culture. As a result, typical payment methods will no longer be used in the future, and therefore they will not be helpful for expanding a business. Customers will not always visit the business with cash in their pockets. They are now placing a premium on debit and credit card payments. As a result, companies will need to update their environment to ensure that they can take all types of payments. In the next years, this situation is expected to become much more severe [1].

In 2020, there were 393,207 cases of CCF out of approximately 1.4 million total reports of identity theft [4]. CCF is now the second most prevalent sort of identity theft recorded as of this year, only following government documents and benefits fraud [5]. In 2020, there were 365,597 incidences of fraud perpetrated using new credit card accounts [10]. The number of identity theft complaints has climbed by 113% from 2019 to 2020, with credit card identity theft reports increasing by 44.6% [14]. Payment card theft cost the global economy $24.26 billion last year. With 38.6% of reported card fraud losses in 2018, the United States is the most vulnerable country to credit theft.

As a result, financial institutions should prioritize equipping themselves with an automated fraud detection system. The goal of supervised CCF detection is to create a machine learning (ML) model based on existing transactional credit card payment data. The model should distinguish between fraudulent and nonfraudulent transactions, and use this information to decide whether an incoming transaction is fraudulent or not. The issue involves a variety of fundamental problems, including the system's quick reaction time, cost sensitivity, and feature pre-processing. ML is a field of artificial intelligence that uses a computer to make predictions based on prior data trends [1]

ML models have been used in many studies to solve numerous challenges. Deep learning (DL) algorithms applied applications in computer network, intrusion detection, banking, insurance, mobile cellular networks, health care fraud detection, medical and malware detection, detection for video surveillance, location tracking, Android malware detection, home automation, and heart disease prediction. We explore the practical application of ML, particularly DL algorithms, to identify credit card thefts in the banking industry in this paper. For data categorisation challenges, the support vector machine (SVM) is a supervised ML technique. It is employed in a variety of domains, including image recognition [25], credit rating [5], and public safety [16]. SVM can tackle linear and nonlinear binary classification problems, and it finds a hyperplane that separates the input data in the support vector, which is superior to other classifiers. Neural networks were the first method used to identify credit card theft in the past [4]. As a result, (DL), a branch of ML, is currently focused on DL approaches.

In recent years, deep learning approaches have received significant attention due to substantial and promising outcomes in various applications, such as computer vision, natural language processing, and voice. However, only a few studies have examined the application of deep neural networks in identifying CCF. [3]. It uses a number of deep learning algorithms for detecting CCF. However, in this study, we choose the CNN model and its layers to determine if the original fraud is the normal transaction of qualified datasets. Some transactions are common in datasets that have been labelled fraudulent and demonstrate questionable transaction behaviour. As a result, we focus on supervised and unsupervised learning in this research paper.

The class imbalance is the problem in ML where the total number of a class of data (positive) is far less than the total number of another class of data (negative). The classification challenge of the unbalanced dataset has been the subject of several studies. An extensive collection of studies can provide several answers. Therefore, to the best of our knowledge, the problem of class imbalance has not yet been solved. We propose to alter the DL algorithm of the CNN model by adding the additional layers for features extraction and the classification of credit card transactions as fraudulent or otherwise. The top attributes from the prepared dataset are ranked using feature selection techniques. After that, CCF is classified using several supervised machine-driven and deep learning models.

In this study, the main aim is to detect fraudulent transactions using credit cards with the help of ML algorithms and deep learning algorithms. This study makes the following contributions:

- Feature selection algorithms are used to rank the top features from the CCF transaction dataset, which help in class label predictions.
- The deep learning model is proposed by adding a number of additional layers that are then used to extract the features and classification from the credit card farad detection dataset.
- To analyse the performance CNN model, apply different architecture of CNN layers.
- To perform a comparative analysis between ML with DL algorithms and proposed CNN with baseline model, the results prove that the proposed approach outperforms existing approaches.
- To assess the accuracy of the classifiers, performance evaluation measures, accuracy, precision, and recall are used. Experiments are performed on the latest credit cards dataset.

The rest of the paper is structured as follows: The second section examines the related works. The proposed model and its methodology are described in depth in Section 3. The dataset and evaluation measures are described in Section 4. It also shows the outcomes of our tests on a real dataset, as well as the analysis. Finally, Section 5 concludes the paper.

II. RELATED WORK

In the field of CCF detection, several research studies have been carried out. This section presents different research studies revolving around CCF detection. Moreover, we strongly emphasise the research that reported fraud detection in the

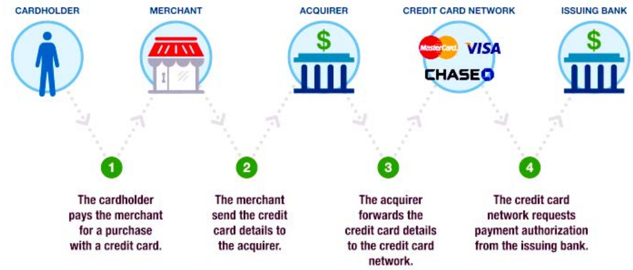

CARDHOLDER
MERCHANT
ACQUIRER
CREDIT CARD NETWORK
ISSUING BANK
VISA
CHASE
1
The cardholder pays the merchant for a purchase with a credit card.
2
The merchant send the credit card details to the acquirer.
3
The acquirer forwards the credit card details to the credit card network.
4
The credit card network requests payment authorization from the issuing bank.

FIGURE 1. **Payment card authorisation process.**

problem of class imbalance. Many techniques are used to detect credit cards. Therefore, to study the most related work in this domain, the main approaches can be categories, such as DL, ML, CCF detection, ensemble and feature ranking, and user authentication approaches [1], [3].

Figure 1 shows the commonly used payment card authorization process for credit card authentication. There are two ways of authentication including passwords and authentication through biometrics. Biometrics-based authentication can be further divided into three groups: physiological authentication and behavioural authentication, and combined authentication [4], [5].

A. SUPERVISED MACHINE LEARNING APPROACHES

ML has many branches, and each branch can deal with different learning tasks. However, ML learning has different framework types. The ML approach provides a solution for CCF, such as random forest (RF). The ensemble of the decision tree is the random forest [3]. Most researchers use the RF approach. To combine the model, we can use (RF) along with network analysis. This method is called APATE [1]. Researchers can use different ML techniques, such as supervised learning and unsupervised techniques. ML algorithms, such as LR, ANN, DT, SVM and NB, are commonly used for CCF detection. The researcher can combine these techniques with ensemble techniques to construct solid detection classifiers [3]. The linking of multiple neurons and nodes is known as an artificial neural network. A feed-forward perceptron multilayer is built up of numerous layers: an input layer, an output layer and one or more hidden layers. For the representation of the exploratory variables, the first layer contains the input nodes. With a precise weight, these input layers are multiplied, and each of the hidden layer nodes is transferred with a certain bias, and they are added together. An activation function is then applied to create the output of each neuron for this summation, which is then transferred to the next layer. Finally, the algorithm's reply is provided by the output layer. The first set randomly used weights and formerly used the training set to minimise the error. All these weights were adjusted by detailed algorithms such as backpropagation [2], [6]. The graphic model for contingency relationships between a set of variables is called the Bayesian belief network. The independence assumption in naïve Bayes is that it was developed to relax and allow for dependencies among variables.

TABLE 1. **Algorithms of machine learning and their accuracy.**

Sr. #	Datasets	Algorithm	Accuracy (%)	Reference
1.	The bankcard enrolment records	LR-based	75	[12]
		RF-based	73	
		GBDT-based	74	
2.	Commercial banks in China	SVM	97.10	[4]
		RF	96.90	
3.	Records of credit card transactions	Light Gradient Boosting Machine algorithm	99.91	[13]
4.	Data are collected in the law enforcement department in China	CS-SVM	98.05	[14]
		GA-SVM	98.05	
		PSO-SVM	98.05	

Variable quantity is characterised as nodes, although dependencies of conditions between variables are shown as arcs between nodes. The conditional probability table of each node is linked, which makes the possibilities of the node's variable conditional on the parent's node values [7], [8]. The computational system of the bilateral-branch network (BBN) is as follows: Finding a construction for the network is the first step: it was raised by human experts, which may be conditional on the specific algorithms by using the data. When this network topology originates, straightforwardly fitting the network uses antique data in naïve Bayes so that the constant variables are also discretised and supposedly distributed normally. Correspondingly, in BBN, it is expected that each node is autonomous of its no offspring, assuming its maternities in the graph [3], [9]. This is acknowledged as the condition of Markov. The linear classification model is a support vector machine (SVM) and problems of regression. Rendering to the SVM algorithm, we can find the points closest to the line from both classes [10], [11]. These points are called support vectors. This paper is concerned with the integration of unsupervised techniques with supervised techniques for the classification of CCF detection. Table 1 presents the summary of machine learning algorithms.

B. DEEP LEARNING APPROACHES

DL algorithms are useful, including the convolutional neural network (CNN) algorithm, and more algorithms are deep belief networks (DBNs) and deep autoencoders; these are considered learning methods. They have numerous layers of processing data, illustration learning and classification of a pattern [7], [15]. The objective of deep-learning is to study artificial neural networks. The standard technique regards the size of neural networks, and it is considered the backpropagation model [8], [16]. The efficiency of the backpropagation algorithm decreases greatly, increasing the depth of the neural networks, which can cause problems, such as insufficient local goals and a dilution of errors. Deep designs should be considered to be an achievement. They can theoretically address the optimisation struggle in a profound manner within the training parameters [17], [18].

The training technique of the deep belief network is often considered the effective primary case of deep architecture training. Traditional ML algorithms, such as SVM, DT and LR, have been extensively proposed for CCF detection [3]. These traditional algorithms are not very well suited for large datasets. A CNN is a DL method; it can deeply relate to three-dimensional data, such as image processing. This method is similar to the ANN; the CNN has the same structure hidden layer and a different number of channels in each layer in addition to special convolution layers. The idea of moving filters through word convolution is linked to the data that can be used to capture the key information and automatically performs feature reduction. Thus, the CNN is widely used in image processing. The CNN does not require heavy data pre-processing for training.

For image processing, the purpose of using a CNN is to minimise processing without losing key features by reducing the image to make predictions [4], [6]. The main terms in the CNN are feature maps, channels, pooling, stride, and padding. For text, image and video processing, CNN models are conventionally used and take two-dimensional data as input, which is called the 2DCNN. To learn the internal representation, the feature mapping process is used from the input data. The location of features is not relevant, and the same procedure can be used for one-dimensional data. Natural language processing is a very popular example of a 1DCNN application where sequence classification becomes a problem. In a 1DCNN, the kernel filter moves top to bottom in a sequence of a data sample, rather than moving left to right and top to bottom in the 2DCNN [17], [18].

Raghavan [16] defined an autoencoder as an actual neural network. An autoencoder can also encrypt the data the same way as it would decrypt the data. In this method, for no anomalous points, the autoencoders are trained. According to the reconstruction error, it would present the anomaly ideas classify it as ’fraud’ or ‘no fraud,’ meaning that the system has not been trained, which is predicted to have a higher amount of anomalies [19], [20]. However, a slight value overhead the higher bound value or considers the threshold an anomaly. This technique is also used in [8], an autoencoder-based network detection of an anomaly. A ML model is a generative adversarial network where two neural networks collaborate to improve their prediction accuracy. GANs are often unsupervised and learn using an obliging zero-sum game framework. The fundamental category of the deep-learning model is a GAN [11], [21], and the perception of development for DL progress it can offer is the most promising direction. GAN takes two main modules. In training, all of the modules make up a model of DL, which is a neural network.

The main two methods used are a generator (G) and a discriminator (D). The network of the generator can generate the data as simulated, and the difference between the simulated data and the target data determines the discriminator, yielding a determination that is true and false around the virtual data. Finally, the model may generate higher-quality simulation data to finish the data creation process [22], [23]. A VAE is a variational autoencoder with regularised training circulation to guarantee that its hidden space has adequate assets, allowing us to create fresh data. A VAE is generated by introducing variation on the basis of the autoencoder. The VEG and the GAN are extremely similar. Once again, the goal is to change and match the data distribution to generate virtual data that is near the target [8], [22].

Usually, the number of samples is similar to that of a normal distribution. If all examples are found, the work can be very successful. Consequently, investigators frequently use neural networks to approximate the mean and modification of normal distribution. Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in DL models [24], [25]. The LSTM network is compatible with categorising, processing and building predictions based on time sequence data. The most common type of RNN is the LSTM. An ordinary neural network (NN) cannot keep track of the preceding information of a learning task every time they have to perform a task. In very simple words, with memory, the RNN is a neural network [26], [27]. RNNs tend to have short-term memory because of the vanishing gradient problem. The backbone of neural networks is backpropagation, as it reduces the loss by weights of network adjustment by using gradients that it originated. In RNNs, as the gradient moves the backbone in the network, it shrinks, and then there is a minor update in weight. These small updates are affected by the earlier layers in the network. They do not learn more, and the RNN loses the ability to recall early examples in long sequences, making it a short-term memory network [28].

The use of DL methods is still very limited, and methods, such as CNN and LSTM are encouraged for image classification, natural language processing (NLP), and RBM because of their ability to handle massive datasets. The way these DL methods perform CCF classification is the major focus of this study [29]. In addition, data pre-processing is an important stage in the ML process. How the classification performance is affected in response to data pre-processing when detecting credit cards is another question that needs to be answered. Table 2 presents the summary of deep learning algorithms.

III. RESEARCH METHODOLOGY

Research is said to be methodical, and research methodology is predicated by the applied research method. Applied research is administered to unravel the issues. Before real-world experimentation, the research covers all fundamentals by performing these steps:

TABLE 2. Accuracy based results of deep learning algorithms.

Sr No	Datasets	Algorithms	Accuracy (%)	Reference
1.	European cards dataset.	LSTM	87.02	[30]
		GRU	86.02	
		Ensemble model ℓ as baseline models	83.37	
2.	The Brazilian dataset.	LSTM	88.47	[30]
		GRU	84.13	
		Ensemble model ℓ as baseline models	79.05	
3.	Commercial banks in China	Deep belief networks (DBN)	97.02	[15]
		CNN	97.24	
		RNN	97.25	
4.	Cardholders Dataset of Europe	GAN	99.95	[31]
		VAE	99.96	

A. LIST OF FEATURES OF CREDIT CARD TRANSACTION DATA

Table 3 lists the important features and shows the mainframe transaction table of credit cards. Even though the whole construction of the transaction information table might be slightly dissimilar amongst card issuers, the vital characteristics recorded would be controlled in the database and are accessible for fraud detection modelling.

1) EXPERIMENTAL STEP-UP

We discuss the dataset to be cast-off and the achievement evaluation measurements to be applied.

a: DESCRIPTION OF DATASET

The credit card dataset is accessible for research purposes. The dataset [11] holds transactions made by a cardholder over a two-day period, i.e., September 2018. There were 284,807 transactions in total, of which 492, or 0.172 percent, were fraudulent. Because disclosing a consumer's transaction details is considered a problem of confidentiality, the main component analysis is applied to the majority of the dataset's features using principal component analysis (PCA). PCA is a standard and widely used technique in the relevant literature for reducing the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss [2], [4], [19]. It does so by creating new uncorrelated variables that successively maximize variance. Table 4 presents the detail of the dataset containing 31 columns, including time, V1, V2, V3...... V28 as PCA applied features, amount, and class labels.

TABLE 3. The list of features available in the CCF dataset.

Sr No.	Name of Feature	Description
1	Account number	Related with account number
2	Open to buy	The availability of balance
3	Credit Limit	The maximum amount of credit of the associated account
4	Card number	Number of Credit card
5	Transaction Amount	The transaction amount submitted by the merchant
6	Transaction Time	Time of the transaction
7	Transaction Date	Date of the transaction
8	Transaction Type	Types of Transaction, such as a cash withdrawal and purchase
9	Currency Code	The currency code
10	Merchant Category Code	The Merchant business type code
11	Merchant Number	The merchant reference number
12	Transaction Country	The country where the transaction takes place
13	Transaction City	The city where the transaction takes place
14	Approval Code	The response to the authorisation request, it means approve or reject.

TABLE 4. Characteristics of the dataset.

S. No	Feature	Description
1.	Time	Time in seconds to require the lapses between the current transaction and the first transaction
2.	V1, V2, V3……V28 attributes	These 28 columns show result of a PCA dimensionality reduction to protect user identities and sensitive features.
3.	Amount	Amount of transaction
4.	Class label	Binary class labels 1 and 0 for nonfraudulent and fraudulent

b: APPLIED MACHINE LEARNING & ENSEMBLE LEARNING TECHNIQUES

We use and apply the following machine and ensemble learning algorithm.

i) EXTREME LEARNING METHOD

The extreme learning method (ELM) is a neural network for classification, clustering, regression and feature learning. It can be used with one or a multilayer of unseen notes. Parameters of unseen nodes are tuned. The weights of the output are hidden nodes learned in a single step. This is the essential amount that is needed to properly learn a linear model. Given a single hidden layer of ELM, we assume that the output function of the j-unseen node is h(z) = G (p, q, z) wherever the parameters of the j^{th} node are. The output function is as follows:

$$f_L(z) = \sum\nolimits_{j=1}^{n} \gamma_i h_i(z) \tag{1}$$

γ_i Is the weight of the output the i^{th} hidden node?

$$\mathrm{h}(z) = |Gh_i(z), \ldots\ldots, h_L(z)| \tag{2}$$

ii) DECISION TREE

As a result, the decision tree classifier is used to create the model, starting with the decision tree. We set the ‘max depth’ to ’4’ in the algorithm, which indicates that the tree can split four times, and the ‘criterion’ to ‘entropy,’ which is similar to ‘max depth’ but decides when to stop splitting the tree. We have thus finished installing and storing everything.

iii) K-NEAREST NEIGHBOURS (KNN)

Supervised Learning is the learning that the amount or the result that we want or expect inside the training data (labelled data), and the amount in the data that we need to learn is known as the Target or the Dependent Variable. Next, for the K-Nearest Neighbours (KNN), we build the model using the ‘K-Neighbours Classifier’ model and take the value of k, which represents the nearest neighbour, as ‘5’. The value of the ‘n-neighbours’ is arbitrarily selected, but it can be selected positively through iterating a range of values, surveyed by fitting and storing the predicted values into the ’knn-yhat’ variable.

iv) RANDOM FOREST (RF)

RF is an ensemble technique and is considered group learning for classifying elements and regression. Deep trees are used to learn irregular patterns. If deep trees learn the same part of the training sample, RF takes an average of its value’s variation, which can be reduced by this method. The training data (p = p1.......pn) with responses (Q = q1, . . . , qn) and bagging (X times) choose a random sample and replace it with the training set that fits the trees for these samples as follows:

For x = 1. . . , X:

$$\frac{1}{X}\sum_{x=1}^{x} f_x(\dot{R}) \tag{3}$$

v) SUPPORT VECTOR MACHINE (SVM)

The SVM algorithm texts effectively. The SVM separates positive and negative instances with high margins. The SVM provides better results than the naïve bayes in earlier studies regarding fraud detection. A decision surface is used to split training points into two categories based on support vectors. Optimisation is calculated as follows:

$$\vec{\alpha} = argmin\left\{-\sum_{j=1}^{n}\alpha_j\sum_{k=1}^{p}\sum_{k=1}^{p}\alpha_i\alpha y_i y\left(\vec{z}_j, \vec{z}_k\right)\right\} \tag{4}$$

$$\sum_{j=1}^{n}\alpha_i y_i = 0; \quad 0 \le \alpha \le \mathrm{C} \tag{5}$$

vi) LOGISTIC REGRESSION

Logistic regression is an easy algorithm that estimates the association between one dependent binary variable and independent variables, computing the probability of the occurrence of an event. The regulation parameter C controls the trade-off between increasing complexity (overfitting) and keeping the model simple (underfitting). For large values of C, the power of regulation is reduced, and the model increases its complexity, thus overfitting the data. The parameter ‘C’ is tuned using Randomised Search CV () for the different datasets: the original, the standardised and the dataset with the most important features. Once the parameter ‘C’ is defined for each dataset, the logistic regression model is initiated and then fitted to the training data, as described in the methodology. The logistic regression hypothesis function can be seen below, where the function $g(z)$ is also shown as follows:

$$h_\theta(x) = g\left(\theta^T x\right) \tag{6}$$

The logistic Regression for the hypothesis can be seen as follows:

$$h(x:) = \frac{1}{1 + e - \theta Tx} \tag{7}$$

Here θ (theta) is a vector of restrictions that our model calculates to appropriate to our classifier.

vii) XG BOOST

The decision-tree-based ensemble ML algorithm is XG Boost,and it uses a framework for gradient boosting. Therefore, when using unstructured data with prediction problems (text, etc.), artificial neural networks tend to outperform all other algorithms or frameworks. The XG boost model for classification is called the XGB Classifier. It can be fit into our training dataset. Models are fit using the sci-kit-learn API and the model’s fit () function. Parameters for training the model can be passed to the model in the constructor. Now, we use serviceable defaults.

c: APPLIED DEEP LEANING TECHNIQUES

We use and apply the following deep learning algorithm.

i) BASELINE MODEL

Essentially, a baseline is a model that has a reasonable chance of providing acceptable results and is simple to set up, usually rapidly experimenting with them, and implementations are widely available in popular packages with low costs.

Classification on Imbalanced Data: This model determines how to classify an extremely imbalanced dataset where the number of examples in one class greatly outnumbers the examples in another.

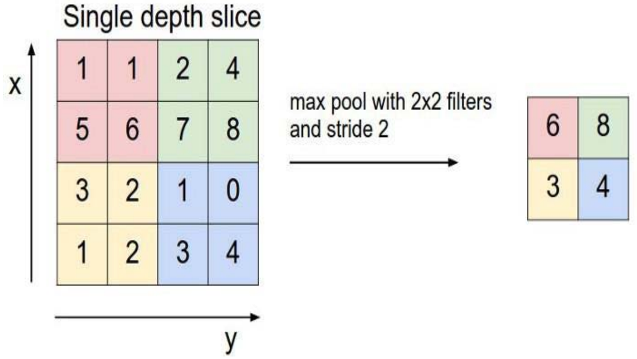


FIGURE 2. Pooling layer.

ii) CONVOLUTIONAL NEURAL NETWORK (CNN)

CNNs, also acknowledged as Conv-Nets, contain multiple layers and are mostly used for processing images. Object detection is widely used for image processing and classification, estimating time series and detecting differences.

Layers in the CNN Model: Here are six distinct layers in the CNN model:

1) Input layer
2) Convo layer (Convo + ReLU)
3) Pooling layer
4) Fully connected layer (FC)
5) SoftMax/logistic layer
6) Output layer

Input Layer: The input layer in the CNN model incorporates CSV data. Text data is characterised by three-dimensional matrices, which should be reshaped into one column.

Convo Layer: The convo layer is occasionally known as the feature extraction layer since the text features are extracted within this layer. First, a part of the text is associated with the Convo layer to make a convolution operation and calculate the dot product between the approachable field and filter. The outcome of the process is a single number of output capacities. The Convo layer also holds the ReLU activation function to build all negative values to zero.

Pooling Layer: The pooling layer is used to decrease the spatial capacity of the input text after convolution. The layer can use two layers of convolution. If we put a fully connected layer after the Convo layer without first including a pooling or max pooling layer, then it will be computationally expensive, which we do not want. Therefore, max pooling must be used to reduce the spatial volume of the input text, as shown in Figure 2.

Fully Connected Layer (FC): A fully connected layer includes weights, biases, and neurons. It attaches the neurons in one layer to the neurons in an additional layer. This layer is used to classify data between dissimilar categories by training.

These categories are:

- *Flattening*
- *Dropout*

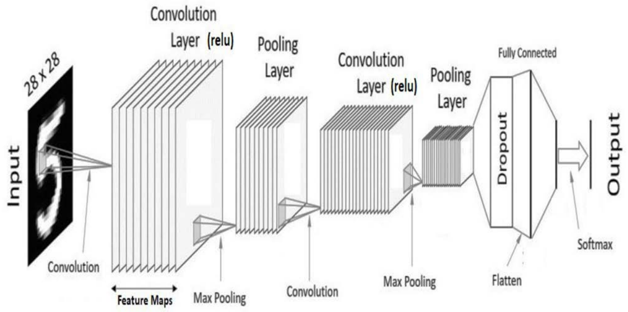


FIGURE 3. CNN output layer.

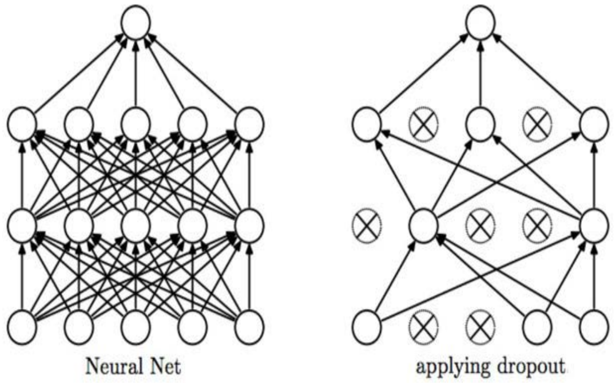


FIGURE 4. Application of dropout over neural network.

SoftMax/Logistic Layer: The SoftMax or Logistic layer is the final layer of the CNN. It is placed after the FC layer and is used for binary classification. Logistic is used, and SoftMax is used for multiclassification.

Output Layer: The output layer holds the label, which is in the procedure of one-hot encoding. Hence, we have a better understanding of CNN. We implement a CNN in Keras. Figure 3 depicts the architecture of CNN from input to output layer.

iii) IMPLEMENTATION WITH KERAS

Creation of the Model: The pipeline of CNN model over keras includes conv layer, max pooling layer, dropout layer, conv layer, max pooling layer, dropout layer along with two fully connected layers sequentially. Figure 4 depicts input neural network and output of dropout layer.

Compile the Model: Categorical Cross-Entropy: We build binary cross-entropy at prior portions and in ML. At that time, we used definite cross-entropy. This means that we have multi-classes. The equation can be seen as follows:

$$CCE = -1/\mathrm{N}\sum_{i=0}^{N}\left(y_j.log(y_j) + \left(1-y_j\right).log(1-y_j)\right) \tag{8}$$

Epochs and Batch Size: We used a dataset of 20 samples, a **batch size** of 2 and determined that the algorithm needed to run for three **epochs**. Consequently, in all epochs, we use five **batches** (20/2 = 10). All batches are run through the algorithm; then, we have five iterations **per epoch**. This method is often an improvement over the sequential model. The most modification comes from the Stalk group and a few slight changes within the module of the sequential model.

d: PERFORMANCE-EVALUTION MEASURES

Traditional methods of estimating ML classifiers can use confusion metrics relating to the difference between the rock bottom dataset truth and the model's prediction where TP, TN, FP, and FN denote true positive, true negative, false-positive and false negative, respectively.

i) ACCURACY

Accuracy is used to measure the performance in the evidence domain recovery and processing of the data. The fraction of the results that are successfully classified can be represented by equation (9) as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

ii) PRECISION

Precision is a performance assessment that measures the ratio of correctly identified positives and the total number of identified positives. This can be seen as follows:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

iii) F-MEASURE/F1-SCORE

The f-measure considers both the precision and the recall. The f-measure may be assumed to be the average weight of all values, which can be seen as follows:

$$F = \frac{2X\ precision\ \times\ Recall}{precision + Recall} \tag{11}$$

iv) RECALL

The recall is also referred to as the sensitivity, which is the ratio of connected instances retrieved over the total number of retrieved instances and can be seen as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

IV. RESULTS AND DISCUSSIONS

A. DATA VISUALISATION

The dataset covers credit cards transactions in October 2018 by European cardholders. The dataset includes transactions that happened in two days, and it includes 492 frauds out of 284,807 transactions. It covers only mathematical input variables, which are the outcome of a PCA transformation. Due to the issue of concealment, we cannot offer the structures of the original dataset and the data more background information. The feature 'Time' covers the seconds elapsed between the first transaction in the dataset and each transaction. Figure 5 shows the class distribution of the CCF dataset into a fraudulent and nonfraud transactions.

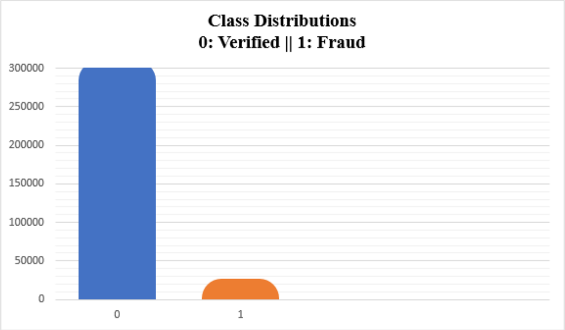


FIGURE 5. Class distribution of fraudulent and nonfraud transactions.

Another insight about the data is that there are no null values; hence, there is no need to fill in missing values.

B. TOP 10 ALGORITHMS IN MACHINE LEARNING FOR FRAUD DETECTION

In the study [3], the top ten ML algorithms are incorporated for the detection of credit card frauds. The list of these algorithms is given below:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. SVM
5. Naïve Bayes
6. CNN
7. K-Means
8. Random Forest
9. Dimensionality Reduction Algorithms
10. Gradient Boosting Algorithms

These algorithms can also encompass association analysis, clustering, classification, statistical learning, and link mining. This is among all the critical topics covered by ML research and development.

1) THE CONFUSION METRICS FOR MODELS

A classification model visualisation is a confusion metric that displays how fit the model is projected to be to the results once associated with the earliest ones. Frequently, the anticipated results are deposited in a variable that is then changed into an association table. Utilizing the association table in the form of a heatmap, the confusion metrics can be plotted. Even though there are numerous built-in methods to envision confusion metrics, we can define and visualize them based on the score to allow for better correlation. Figure 6 depicts the confusion metrics of machine learning algorithms.

2) THE ACCURACY OF MACHINE LEARNING ALGORITHMS

In this phase, we structure six distinct kinds of classification models. We could use numerous other models to resolve classification problems; however, these are the most popular models in use. Using the algorithms, all these models can be built workably provided by the sci-kit-learn package. The results of applied ML algorithms are presented in Table 5.

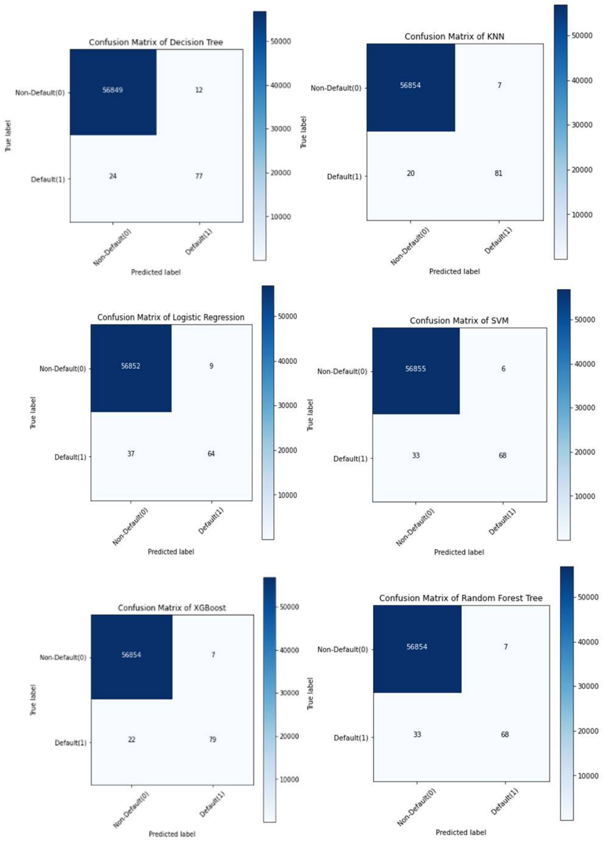


FIGURE 6. Confusion metrics of machine learning algorithms.

TABLE 5. The accuracy and F1-socre of machine learning algorithms.

Sr No	Algorithm Name	Accuracy Score (%)	F1 Score (%)
1.	Decision tree algorithm	99.93	81.05
2.	KNN algorithm	99.95	85.71
3.	Logistic regression algorithm	99.91	73.56
4.	SVM Algorithms	99.93	77.71
5.	Random forest tree algorithm	99.92	77.27
6.	XG Boost	99.94	84.49

3) RESULT OF THE CASE AMOUNT STATISTICS OF THE DATASET

As shown in Figure 7, the case count statistics, the values of the ’Amount’ variable vary substantially once associated with the respite of the variables. To decrease the wide range of the values, we can standardise it by means of the *‘Standard-Scaler’* method in Python.

4) THE COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS

Figure 8 show the comparative analysis of applied ML algorithms for CCF using accuracy and F1 measure metrics.

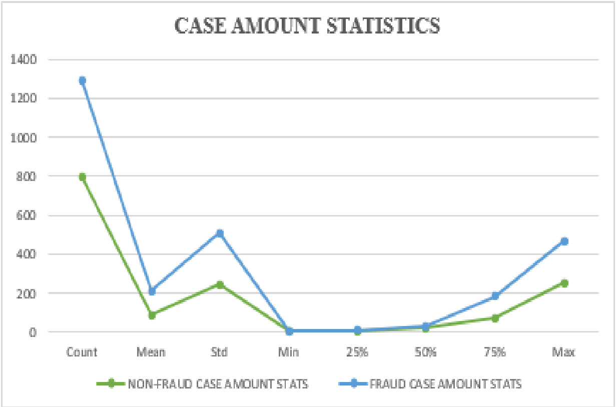


FIGURE 7. The case count statistics for fraud and non-fraud transactions.

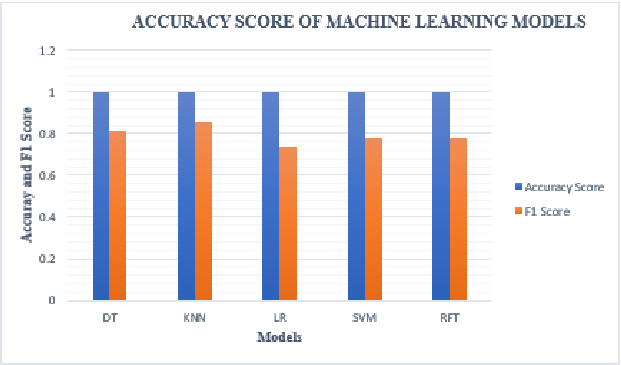


FIGURE 8. Comparative analysis of machine learning algorithms.

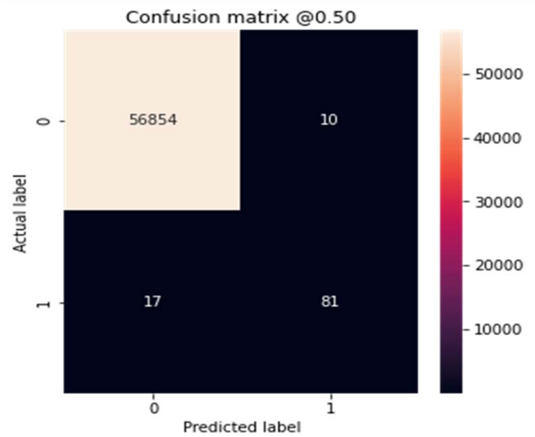


FIGURE 9. Metrics of deep learning with epoch sizes as 35 and 14.

C. TOP 10 ALGORITHMS IN DEEP LEARNING FOR FRAUD DETECTION

In [8], ten DL algorithms are identified as top algorithms d. The list of these algorithms is given below:

1. Convolutional neural networks (CNN)
2. Long short-term memory (LSTM)
3. Residual neural network (RNN)

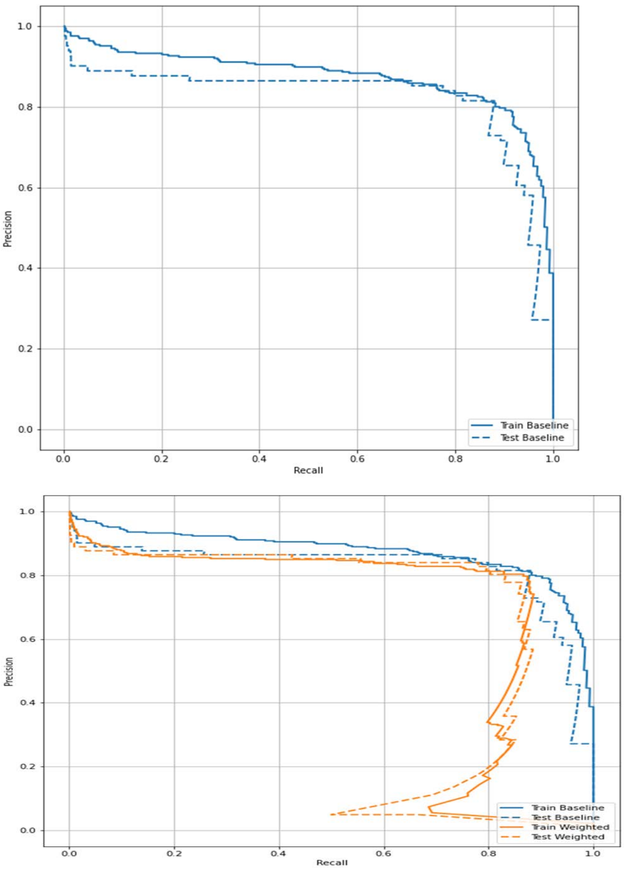


FIGURE 10. Area under the interpolated precision-recall curve.

4. Baseline (BL)
5. Generative adversarial networks (GAN)
6. Radial basis function network (RBFN)
7. Multilayer perception (MLP)
8. Self-organise map (SOM)
9. Deep belief network (DBN)
10. Restricted Boltzmann machine (RBM)
11. Autoencoders

1) THE EVALUATION METRICS

We can use confusion metrics to summarise the labels of actual vs. predicted, wherever the X-axis is the label of the predicted, and the Y-axis is the label of the actual:

If the model had projected the whole thing accurately, this would be a diagonal metric whose values would be away from the main diagonal and demonstrate an incorrect prediction value of zero. In this case, the metrics display that because of the comparatively rare false-positives, it is determined that a few legitimate transactions were flagged incorrectly. This trade-off might be desirable because false negatives would permit more fraudulent transactions to go through.

2) THE ACCURACY OF DEEP LEARNING ALGORITHMS

Table 7 shows the training and validation accuracy of proposed CNN and baseline CNN algorithms. The CNN model is applied by varying the layers from 11 to 20 and comparing the result with baseline 5-layer architecture.

TABLE 6. The result of CNN model using epoch size as 35 and 14.

Metrics	Epoch Size 35	Epoch Size 14
Loss	0.004	0.014
TP	83.0	87
FP	6.0	66
TN	56849.0	56789
FN	24.0	20
Accuracy	0.999	0.998
Precision	0.932	0.568
Recall	0.775	0.813
AUC	0.929	0.942
PRC	0.816	0.741
Total Fraudulent Transactions	**107**	**107**

TABLE 7. The accuracy of deep learning models using different epochs.

Name of Algorithms	Layers	Epoch Size	Training Accuracy	Validation Accuracy
CNN with imbalanced	11	20	0.932	0.913
	13	50	0.912	0.913
	14	100	0.963	0.943
	17	100	0.955	0.947
	20	100	0.949	0.935
CNN with balanced dataset	14	100	0.946	0.958
Baseline	05	20	0.907	0.883

3) THE SUMMARY OF THE CNN MODEL

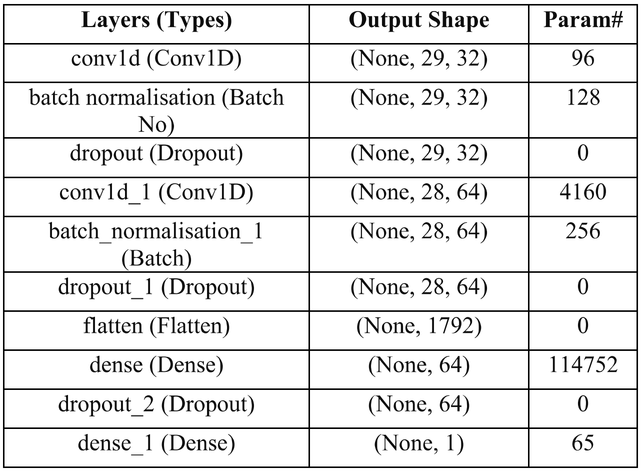
Once a model is "built", the summary () method can be called to show its details as shown in Table 8. However, it can be beneficial when constructing a sequential model incrementally to show the summary of the model thus far with the current output.

The total number of parameters is 119,457 and the total number of trainable parameters is 119,265. Finally, the number of nontrainable parameters is 192.

4) THE SUMMARY OF THE BASELINE MODEL

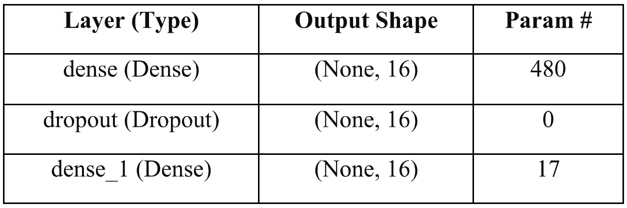
By using the function, we now develop and train the previously defined model. Note that the model is best suited to using a batch size larger than 2048; this is important for confirming that each batch has a decent chance of comprising

TABLE 8. The summary of CNN sequential model.



Layers (Types)	Output Shape	Param#
conv1d (Conv1D)	(None, 29, 32)	96
batch normalisation (Batch No)	(None, 29, 32)	128
dropout (Dropout)	(None, 29, 32)	0
conv1d_1 (Conv1D)	(None, 28, 64)	4160
batch_normalisation_1 (Batch)	(None, 28, 64)	256
dropout_1 (Dropout)	(None, 28, 64)	0
flatten (Flatten)	(None, 1792)	0
dense (Dense)	(None, 64)	114752
dropout_2 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

TABLE 9. The summary of baseline CNN sequential model.



Layer (Type)	Output Shape	Param #
dense (Dense)	(None, 16)	480
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 16)	17

a rare positive fraud example. The summary of the baseline model is presented in Table 9.

The total amount of parameters is 497 and the total number of trainable parameters is 497. Finally, the total amount of nontrainable parameters is 0.

5) DISTRIBUTION OF THE DATA

Identifying fraudulent credit card transactions is a common type of imbalanced binary classification where the focus is on the positive class (is fraud) class and negative class (is not fraud) class. Then, we compare the classification of the positive and negative instances over a rare feature. The positive and negative distributions are shown in Figure 11 and Figure 12 respectively.

6) VARIATION OF EPOCHS

We train the model for 20 and 30 epochs, with and without careful initialisation, and compare the losses. The figure clearly shows that careful initialisation gives a clear advantage in regard to validation loss. Figure 13 shows the validation loss using zero bias and careful bias.

7) RECORD OF THE TRAINING DATASET

In this section, we construct schemes of the model's accuracy and loss on the training and validation sets. We check for overfitting; these measurements are valuable too, as they can help us learn more about the overfitting and underfitting of the model. Figure 14 depicts the training and validation loss, precision recall accuracy (prc), precisions and recall over 35 epochs.

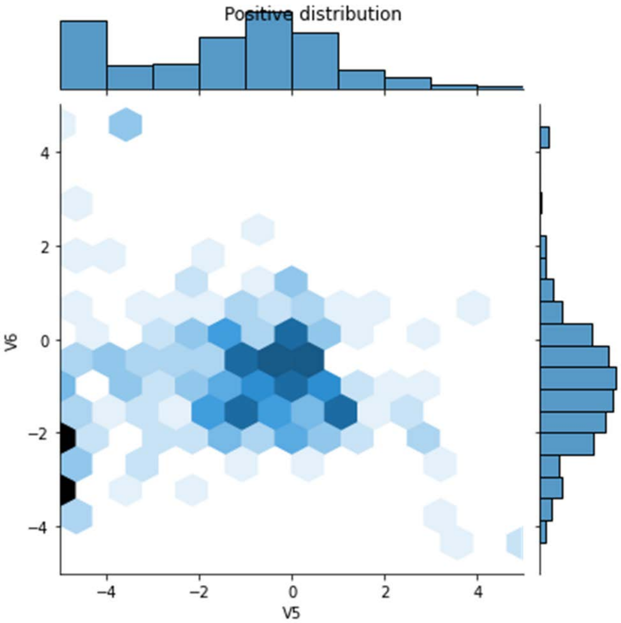


FIGURE 11. Positive distribution of the data.

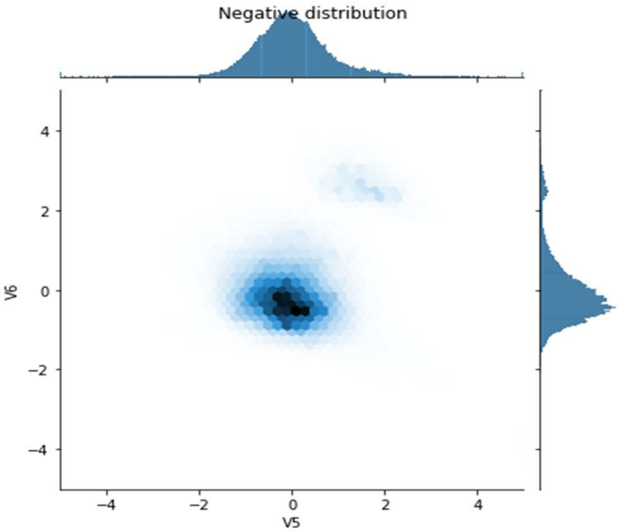


FIGURE 12. Negative distribution of the data.

Table 10 presents the training and validation results of baseline deep learning model using 35 and 14 epochs.

8) THE DIAGNOSIS MODEL BEHAVIOUR

The behaviour of a ML and DL model can be used to diagnose the shape and dynamics of a learning curve and to possibly recommend the best configuration changes for improving performance and learning. There are four learning curves: Underfit, Overfit, Good Fit, Epoch. The learning curve is used to plot the model for training and validation accuracy and

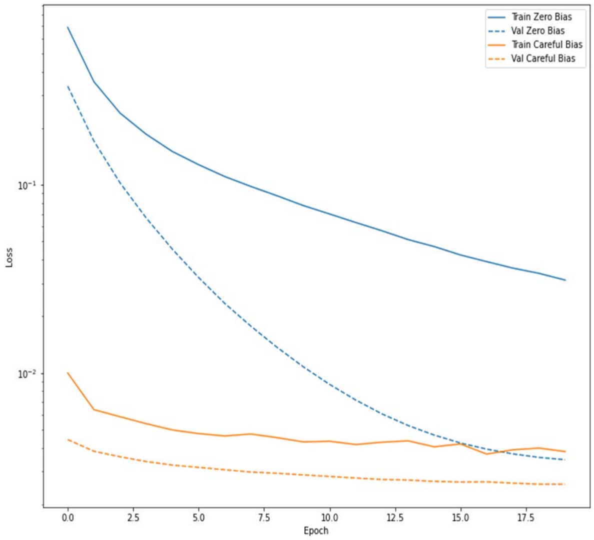


FIGURE 13. **Validation loss using zero bias and careful bias.**

TABLE 10. **Results of deep learning model using different epochs.**

Baseline Model	Epoch 35		Epoch 14	
Metrics	Training Accuracy (%)	Validation Accuracy (%)	Training Accuracy (%)	Validation Accuracy (%)
Precision	93	42	91	89
Recall	90	85	80	68
AUC	98	97	94	95
PRC	56	22	84	80
Accuracy	98	96	99	99

training and validation loss vs. epochs. We display overfitting over the epochs, which is where validation accuracy is less than training accuracy and epochs where validation loss is greater than the training loss.

9) RESULTS OF DL ALGORITHMS ON BALANCED DATA

The imbalanced CCF dataset is transformed into a balanced dataset by removing non fraudulent transactions from the dataset. In a real-world transaction, fraudulent and non-fraudulent classes are not balanced due to the nature of the problem. For instance, if one million transactions are performed in a day, only a few can be fraudulent. The convolutional neural network model with 14 layers architecture is applied to the balanced dataset to validate the proposed model. The model is trained over 100 epochs. The CNN 14 layers architecture obtained 94.60 and 95.80 % training and validation accuracy respectively as shown in Table 7. Figure 15 depicts the accuracy and loss of CNN model using the balanced CCF dataset.

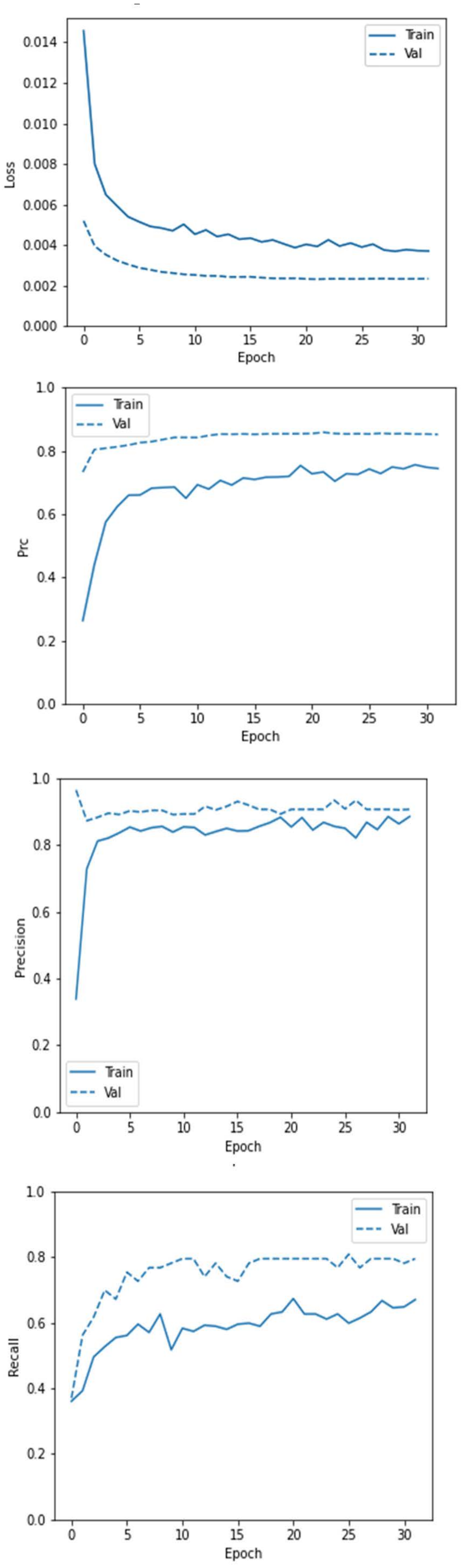


FIGURE 14. **Training and validation history of loss, precision Recall Accuracy (PRC), precisions and recall (Epoch size 35).**

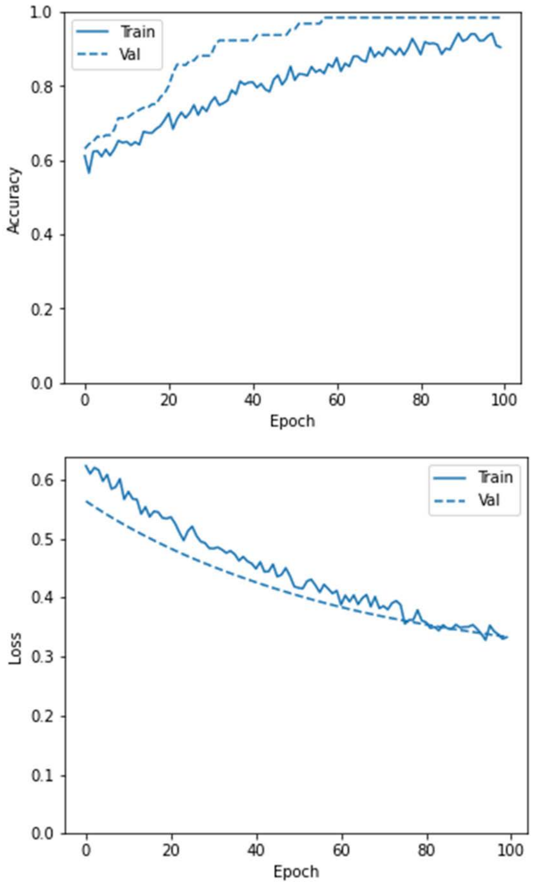


FIGURE 15. **Training and validation history of accuracy and loss of CNN model using 100 epochs.**

10) PLOT TRAINING & VALIDATION ACCURACY VALUE

Figure 16 depicts the training and validation accuracy of proposed model over 20 and 50 epochs.

11) RESULT OF THE CNN LAYERS IMPLEMENTATION

Our proposed sequential model has a convolutional layer with 32 filters of size 3 and a ReLU activation function, which is followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.25. Figure 17 depicts the accuracy of CNN model using different layers architecture. The architectures of our proposed model are as follows.

a: ARCHITECTURE OF 14 LAYERS

Our proposed model has 14 layers: a convolutional layer with a kernel size of 32 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.2. Then, we add another convolutional layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.5. Then, we add a flattened layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a dense layer and a dropout layer with a dropout rate of 0.5, followed by 3 dense layers. The first dense layer has a ReLU activation function of (100). The second dense has a ReLU activation function of (50). The third dense layer has a ReLU activation function of (25). Finally, we add a dense layer for classification with a sigmoid activation function. At 100 epochs, the accuracy is 96.34%.

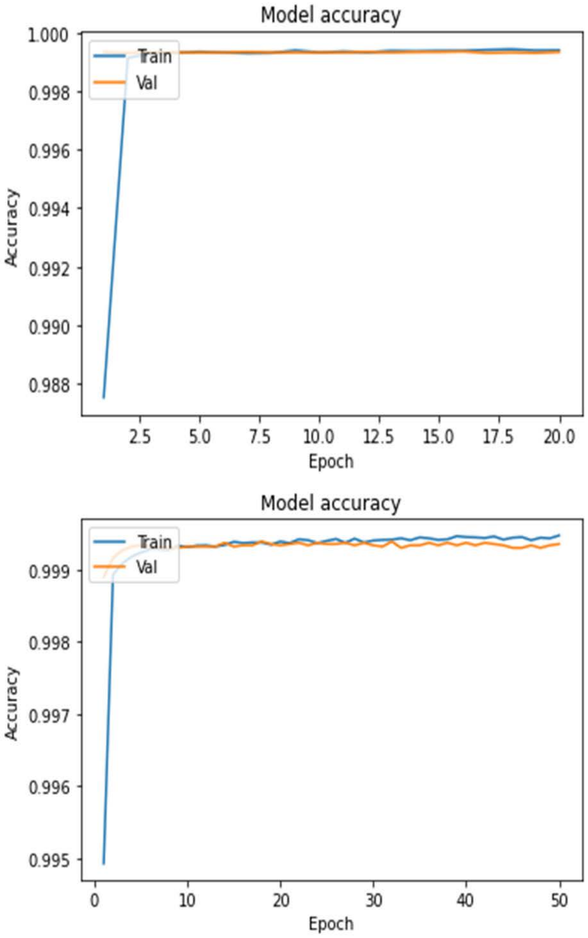
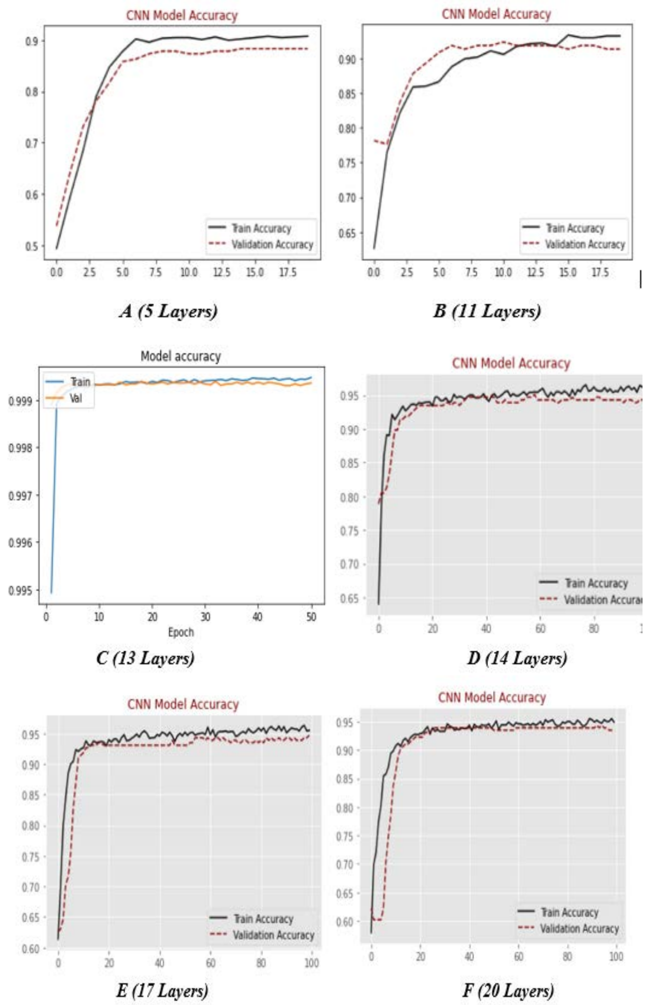


FIGURE 16. **Model accuracy when epoch sizes are 20 and 50.**

b: ARCHITECTURE OF 17 LAYERS

Our proposed model has 17 layers: a convolutional layer with a kernel size of 32 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.2. Then, we add another convolutional layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.5. Then, we add another convolutional layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.25.

Then, we add a flattened layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a dense layer and a dropout layer with a dropout rate of 0.5, followed by 3 dense layers. The first dense layer has a ReLU activation function of (100). The second dense layer has a ReLU activation function of (50). The third dense layer has a ReLU activation function



A (5 Layers)

B (11 Layers)

C (13 Layers)

D (14 Layers)

E (17 Layers)

F (20 Layers)

FIGURE 17. Accuracy of the CNN model over number of layers.

of (25). Finally, we add a dense layer for classification with a sigmoid activation function. After 100 epochs, the accuracy is 95.53%.

c: ARCHITECTURE OF 20 LAYERS

Our proposed model has 20 layers: a convolutional layer with a kernel size of 32 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.2. Then, we add another convolutional layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.5. Then, we add another convolutional layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.5.

Then, we add another convolutional layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a batch normalisation layer and a dropout layer with a dropout rate of 0.25. Then, we add a flattened layer with a kernel size of 64 × 2 and a ReLU activation function, followed by a dense layer and a dropout layer with a dropout rate of 0.5, followed by 3 dense layers. The first dense layer has a ReLU activation function of (100). The second dense layer has a ReLU activation function of (50). The third dense layer has a ReLU activation function of (25). Finally, we add a dense layer for classification with a sigmoid activation function. At 100 epochs, the accuracy is 94.92%.

TABLE 11. Comparative analysis of ML and DL algorithms.

Model Name	Existing Accuracy (%)	New Accuracy (%)	Reference
CNN	93.00 94.6	96.34	[32] [24]
BL	83	99.72	[22]
RF	97.55 92.3	99.92	[32] [24]
SVM	97.43	99.93	[32]
KNN	93.27 91.11	99.91	[33] [34]
DT	97.08 66.5	99.93	[35] [24]
LR	97.18 67.8	99.91	[35] [24]

12) THE COMPARATIVE ANALYSIS OF THE MACHINE LEARNING AND DEEP LEARNING ALGORITHMS

The most important distinction between DL and standard ML is how well deep learning performs when the amount of data changes, as DL techniques do not perform well when the amount of data is minimal. This is because DL algorithms require a large quantity of data to fully learn features. ML algorithms are less accurate than deep learning algorithms. Therefore, the existing accuracy of ML algorithms and DL algorithms is low compared to the accuracy of the proposed model. Table 10 presents a comparative analysis of ML and DL algorithms.

V. CONCLUSION AND FUTURE WORK

CCF is an increasing threat to financial institutions. Fraudsters tend to constantly come up with new fraud methods. A robust classifier can handle the changing nature of fraud. Accurately predicting fraud cases and reducing false-positive cases is the foremost priority of a fraud detection system. The performance of ML methods varies for each individual business case. The type of input data is a dominant factor that drives different ML methods. For detecting CCF, the number of features, number of transactions, and correlation between the features are essential factors in determining the model's performance. DL methods, such as CNNs and their layers, are associated with the processing of text and the baseline model. Using these methods for the detection of credit cards yields better performance than traditional algorithms. Comparing all the algorithm performances side to side, the CNN with 20 layers and the baseline model is the top method with an accuracy of 99.72%. Numerous sampling techniques are used to increase the performance of existing examples, but they significantly decrease on the unseen data. The performance on unseen data increased as the class imbalance increased. Future work associated may explore the use of more state of art deep learning methods to improve the performance of the model proposed in this study.

REFERENCES

[1] Y. Abakarim, M. Lahby, and A. Attioui, "An efficient real time model for credit card fraud detection based on deep learning," in *Proc. 12th Int. Conf. Intell. Systems: Theories Appl.*, Oct. 2018, pp. 1–7, doi: 10.1145/3289402.3289530.

[2] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, Jul. 2010, doi: 10.1002/wics.101.

[3] V. Arora, R. S. Leekha, K. Lee, and A. Kataria, "Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence," *Mobile Inf. Syst.*, vol. 2020, pp. 1–13, Oct. 2020, doi: 10.1155/2020/8885269.

[4] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, "Performance analysis of feature selection methods in software defect prediction: A search method approach," *Appl. Sci.*, vol. 9, no. 13, p. 2764, Jul. 2019, doi: 10.3390/app9132764.

[5] B. Bandaranayake, "Fraud and corruption control at education system level: A case study of the Victorian department of education and early childhood development in Australia," *J. Cases Educ. Leadership*, vol. 17, no. 4, pp. 34–53, Dec. 2014, doi: 10.1177/1555458914549669.

[6] J. Błaszczyński, A. T. de Almeida Filho, A. Matuszyk, M. Szelg, and R. Słowiński, "Auto loan fraud detection using dominance-based rough set approach versus machine learning methods," *Expert Syst. Appl.*, vol. 163, Jan. 2021, Art. no. 113740, doi: 10.1016/j.eswa.2020.113740.

[7] B. Branco, P. Abreu, A. S. Gomes, M. S. C. Almeida, J. T. Ascensão, and P. Bizarro, "Interleaved sequence RNNs for fraud detection," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 3101–3109, doi: 10.1145/3394486.3403361.

[8] F. Cartella, O. Anunciacao, Y. Funabiki, D. Yamaguchi, T. Akishita, and O. Elshocht, "Adversarial attacks for tabular data: Application to fraud detection and imbalanced data," 2021, *arXiv:2101.08030*.

[9] S. S. Lad, I. Dept. of CSERajarambapu Institute of TechnologyRajaramnagarSangliMaharashtra, and A. C. Adamuthe, "Malware classification with improved convolutional neural network model," *Int. J. Comput. Netw. Inf. Secur.*, vol. 12, no. 6, pp. 30–43, Dec. 2021, doi: 10.5815/ijcnis.2020.06.03.

[10] V. N. Dornadula and S. Geetha, "Credit card fraud detection using machine learning algorithms," *Proc. Comput. Sci.*, vol. 165, pp. 631–641, Jan. 2019, doi: 10.1016/j.procs.2020.01.057.

[11] I. Benchaji, S. Douzi, and B. E. Ouahidi, "Credit card fraud detection model based on LSTM recurrent neural networks," *J. Adv. Inf. Technol.*, vol. 12, no. 2, pp. 113–118, 2021, doi: 10.12720/jait.12.2.113-118.

[12] Y. Fang, Y. Zhang, and C. Huang, "Credit card fraud detection based on machine learning," *Comput., Mater. Continua*, vol. 61, no. 1, pp. 185–195, 2019, doi: 10.32604/cmc.2019.06144.

[13] J. Forough and S. Momtazi, "Ensemble of deep sequential models for credit card fraud detection," *Appl. Soft Comput.*, vol. 99, Feb. 2021, Art. no. 106883, doi: 10.1016/j.asoc.2020.106883.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[15] X. Hu, H. Chen, and R. Zhang, "Short paper: Credit card fraud detection using LightGBM with asymmetric error control," in *Proc. 2nd Int. Conf. Artif. Intell. for Industries (AII)*, Sep. 2019, pp. 91–94, doi: 10.1109/AI4I46381.2019.00030.

[16] J. Kim, H.-J. Kim, and H. Kim, "Fraud detection for job placement using hierarchical clusters-based deep neural networks," *Int. J. Speech Technol.*, vol. 49, no. 8, pp. 2842–2861, Aug. 2019, doi: 10.1007/s10489-019-01419-2.

[17] M.-J. Kim and T.-S. Kim, "A neural classifier with fraud density map for effective credit card fraud detection," in *Intelligent Data Engineering and Automated Learning*, vol. 2412, H. Yin, N. Allinson, R. Freeman, J. Keane, and S. Hubbard, Eds. Berlin, Germany: Springer, 2002, pp. 378–383, doi: 10.1007/3-540-45675-9_56.

[18] N. Kousika, G. Vishali, S. Sunandhana, and M. A. Vijay, "Machine learning based fraud analysis and detection system," *J. Phys., Conf.*, vol. 1916, no. 1, May 2021, Art. no. 012115, doi: 10.1088/1742-6596/1916/1/012115.

[19] R. F. Lima and A. Pereira, "Feature selection approaches to fraud detection in e-payment systems," in *E-Commerce and Web Technologies*, vol. 278, D. Bridge and H. Stuckenschmidt, Eds. Springer, 2017, pp. 111–126, doi: 10.1007/978-3-319-53676-7_9.

[20] Y. Lucas and J. Jurgovsky, "Credit card fraud detection using machine learning: A survey," 2020, *arXiv:2010.06479*.

[21] H. Zhou, H.-F. Chai, and M.-L. Qiu, "Fraud detection within bankcard enrollment on mobile device based payment using machine learning," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 12, pp. 1537–1545, Dec. 2018, doi: 10.1631/FITEE.1800580.

[22] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019, doi: 10.1109/ACCESS.2019.2927266.

[23] I. Matloob, S. A. Khan, and H. U. Rahman, "Sequence mining and prediction-based healthcare fraud detection methodology," *IEEE Access*, vol. 8, pp. 143256–143273, 2020, doi: 10.1109/ACCESS.2020.3013962.

[24] I. Mekterović, M. Karan, D. Pintar, and L. Brkić, "Credit card fraud detection in card-not-present transactions: Where to invest?" *Appl. Sci.*, vol. 11, no. 15, p. 6766, Jul. 2021, doi: 10.3390/app11156766.

[25] D. Molina, A. LaTorre, and F. Herrera, "SHADE with iterative local search for large-scale global optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8, doi: 10.1109/CEC.2018.8477755.

[26] M. Muhsin, M. Kardoyo, S. Arief, A. Nurkhin, and H. Pramusinto, "An analyis of student's academic fraud behavior," in *Proc. Int. Conf. Learn. Innov. (ICLI)*, Malang, Indonesia, 2018, pp. 34–38, doi: 10.2991/icli-17.2018.7.

[27] H. Najadat, O. Altiti, A. A. Aqouleh, and M. Younes, "Credit card fraud detection based on machine and deep learning," in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2020, pp. 204–208, doi: 10.1109/ICICS49469.2020.239524.

[28] A. Pumsirirat and L. Yan, "Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 1, pp. 18–25, 2018, doi: 10.14569/IJACSA.2018.090103.

[29] P. Raghavan and N. E. Gayar, "Fraud detection using machine learning and deep learning," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Dec. 2019, pp. 334–339, doi: 10.1109/ICCIKE47802.2019.9004231.

[30] M. Ramzan, A. Abid, H. U. Khan, S. M. Awan, A. Ismail, M. Ahmed, M. Ilyas, and A. Mahmood, "A review on State-of-the-Art violence detection techniques," *IEEE Access*, vol. 7, pp. 107560–107575, 2019, doi: 10.1109/ACCESS.2019.2932114.

[31] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas, and A. Mahmood, "A survey on state-of-the-art drowsiness detection techniques," *IEEE Access*, vol. 7, pp. 61904–61919, 2019, doi: 10.1109/ACCESS.2019.2914373.

[32] A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," *Global Transitions Proc.*, vol. 2, no. 1, pp. 35–41, Jun. 2021, doi: 10.1016/j.gltp.2021.01.006.

[33] N. F. Ryman-Tubb, P. Krause, and W. Garn, "How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark," *Eng. Appl. Artif. Intell.*, vol. 76, pp. 130–157, Nov. 2018, doi: 10.1016/j.engappai.2018.07.008.

[34] I. Sadgali, N. Sael, and F. Benabbou, "Adaptive model for credit card fraud detection," *Int. J. Interact. Mobile Technol.*, vol. 14, no. 3, p. 54, Feb. 2020, doi: 10.3991/ijim.v14i03.11763.

[35] Y. Sahin and E. Duman, "Detecting credit card fraud by ANN and logistic regression," in *Proc. Int. Symp. Innov. Intell. Syst. Appl.*, Jun. 2011, pp. 315–319, doi: 10.1109/INISTA.2011.5946108.

[36] I. Sohony, R. Pratap, and U. Nambiar, "Ensemble learning for credit card fraud detection," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Jan. 2018, pp. 289–294, doi: 10.1145/3152494.3156815.

[37] B. Stojanović, J. Božić, K. Hofer-Schmitz, K. Nahrgang, A. Weber, A. Badii, M. Sundaram, E. Jordan, and J. Runevic, "Follow the trail: Machine learning for fraud detection in fintech applications," *Sensors*, vol. 21, no. 5, p. 1594, Feb. 2021, doi: 10.3390/s21051594.

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.

[39] H. Tingfei, C. Guangquan, and H. Kuihua, "Using variational auto encoding in credit card fraud detection," *IEEE Access*, vol. 8, pp. 149841–149853, 2020, doi: 10.1109/ACCESS.2020.3015600.

[40] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection–machine learning methods," in *Proc. 18th Int. Symp. INFOTEH-JAHORINA (INFOTEH)*, Mar. 2019, pp. 1–5, doi: 10.1109/INFOTEH.2019.8717766.

[41] S. Warghade, S. Desai, and V. Patil, "Credit card fraud detection from imbalanced dataset using machine learning algorithm," *Int. J. Comput. Trends Technol.*, vol. 68, no. 3, pp. 22–28, Mar. 2020, doi: 10.14445/22312803/IJCTT-V68I3P105.

[42] N. Yousefi, M. Alaghband, and I. Garibay, "A comprehensive survey on machine learning techniques and user authentication approaches for credit card fraud detection," 2019, *arXiv:1912.02629*.

[43] X. Zhang, Y. Han, W. Xu, and Q. Wang, "HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," *Inf. Sci.*, vol. 557, pp. 302–316, May 2021, doi: 10.1016/j.ins.2019.05.023.

FAWAZ KHALED ALARFAJ received the M.Sc. and Ph.D. degrees in computer science from Essex University, U.K. He is currently an Assistant Professor with the Computer and Information Sciences Department, Imam Muhammad Ibn Saud Islamic University (IMSIU). His research interests include information retrieval, natural language processing, machine learning, big data, and cloud computing.

IQRA MALIK is currently pursuing the master's degree in computer science with the Department of Computer Science and Information Technology, University of Sargodha, Sargodha, Pakistan. She is also a Research Scholar with the Department of Computer Science and Information Technology, University of Sargodha. Her research interests include machine learning, deep learning, digital image processing, and computer vision.

HIKMAT ULLAH KHAN received the master's and Ph.D. degrees in computer science from International Islamic University, Islamabad. He has been an Active Researcher for the last ten years. He is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Wah Cantt, Pakistan. He has authored more than 50 papers in top peer-reviewed journals and international conferences. His research interests include social web mining, semantic web, data science, information retrieval, and scientometrics. He is an editorial board member of a number of prestigious impact factor journals.

NAIF ALMUSALLAM received the B.S. degree in computer science from King Faisal University, Hofuf, Saudi Arabia, in 2009, the M.S. degree from Monash University, Melbourne, VIC, Australia, in 2013, and the Ph.D. degree in computer science from RMIT University, Melbourne, in 2019. He is currently an Assistant Professor with Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. His research interests include machine learning, data science, and security.

MUHAMMAD RAMZAN is currently pursuing the Ph.D. degree with the University of Management and Technology, Lahore, Pakistan.

He is also a Lecturer with the University of Sargodha, Pakistan. He has authored several research articles published in well reputed peer-reviewed journals. His research interests include algorithms, machine learning, software engineering, and computer vision.

MUZAMIL AHMED received the M.S. degree in computer science from the University of Lahore, Pakistan. He is currently pursuing the Ph.D. degree with the Department of Computer Science, COMSATS University Islamabad, Wah Cantt, Pakistan. His research interests include natural language processing, machine learning, deep learning, data science, information retrieval, and digital image processing.