Financial Fraud Detection Using Deep Learning Techniques

1st Agolapu Dileep
Department of CSE
Vardhaman College of Engineering
Hyderabad, India
dileepagolapugoud@gmail.com

2nd Amma Karthik
Department of CSE
Vardhaman College of Engineering
Hyderabad, India
ammakarthikedu@gmail.com

3rd Gujja Shiva Krishna
Department of CSE
Vardhaman College of Engineering
Hyderabad, India
gujjashivakrishna3@gmail.com

4th D Ganesh
Department of CSE
Vardhaman College of Engineering
Hyderabad, India
dganesh@vardhaman.org

5th Shanmugasundaram Hariharan
Department of CSE
Vardhaman College of Engineering
Hyderabad, India
mailtos.hariharan@gmail.com

2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) | 979-8-3503-4745-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDCECE57866.2023.10150467

***Abstract*—We live in a modern world where people often find it easy and are merely to go with the cashless payments. This advancement in technology is very helpful but ill effects have its ill effects as well. Due to these card payments, privacy can be a myth. Because there is a gap for fraud to take place. If a fraud takes place, it cannot be detected as there are a lot of normal transactions incoming. Segregation of the fraudulent ones from the normal transactions is the real difficulty because the fraudulent transactions are very less in number when compared to the normal transactions. A lot of techniques have been used by researchers so far. This research work focuses to detect the fewest methods with fewer false positives and minimal false negatives, to correctly classify fraudulent transactions by working on a variety of Machine learning and Deep learning techniques. The Dataset that we have chosen are namely, FraudTrain and FraudTest. Their respective data samples are 1296675 and 555719. Unsupervised ML techniques like Local Outlier Factor (LOF), IsolationvForest (IF) and Neural Networks like Autoencoders were used. In our experiment, LOF has outperformed with 99.0% accuracy, 0.8 recall, followed by Autoencoders with 96.40 accuracy, 0.22 recall, and Isolation Forest with 98.92% accuracy and 0.01 recall score.**

Keywords—Deep Learning, Autoencoders, Local Outlier Factor, isolation Forest.

I. Introduction

Financial Fraud Detection refers to the criminal activities that occur in a commercial organization, which leads to the monetary loss as well damage to the reputation of a company. Damage to reputation eventually not only leads to the loss of customers but also their trust in that organization. Hence, we can say that financial fraud can worst affect an organization in all the possible ways it can. Therefore, the organizations which come under the financial sector like banks, credit card companies, stock markets, and insurance companies should meticulously look for malicious users who try to take advantage of the vulnerabilities [13].

Detecting malicious users is a nightmare for any company as those malicious users can be the employees within that organization or the customers of that company or a fraudulent user masquerading as a regular customer. These fraudulent users try to make use of the company's resources in an illegal way. This is how fraud occurs and it incurs a huge economic loss to the companies. Hence the companies are very keen and are interested in immediate defrauding of this sort of frauds. That is why there came many techniques to detect such frauds [4].

One approach is Activity Monitoring, in which a custom tire activity is before monitored virtual logged after in the customer has logged in and till the end of the session. Meanwhile, if a customer is seen behaving abnormally then he/she is being flagged and an issue or alert is raised. Apart from this, there's another approach in which the profile of each customer is maintained. A Customer Profile consists of a usage pattern of his transactions, with which he/she is being monitored and any abnormal behavior or deviation is being flagged and an alert is raised [6].

This research paper's material is organized as follows: The literature review is included in Section II, the methodology is covered in Section III, findings are covered in Section IV, and the conclusion is covered in Section V.

II. Literature Review

Different techniques that are used by authors in 2018 proposed a study that uses an un-supervised fraud detection approach for detecting fraud with the help of Autoencoder based clustering, which is a combination of an autoencoder with 3 hidden layers and a K-Means Clustering algorithm. Activation functions like elu, relu were used with RMSProp optimizer having 0.1 learning rate. This model has been implemented using that set which consists of 284807 transactions. An accuracy of 98.9% and TPR of 81% is achieved by this model. This shows us that the model clearly outperforms others. But the only disadvantage of this model is its highest computational cost [1].

They have conducted a survey on various ML and DL techniques like Neural Networks, Isolation Forest, Local-Outlier Factor, KNN, SVM, and Logistic Regression etc. This model works on the basis of distance sum rule to detect the outliers. The dataset used has undergone PCA transformation earlier to make the data confidential, which consists of 2,84,807 legitimate transactions and is highly imbalanced. Among all, Isolation Forest worked very well and gave an accuracy of 99.74% followed by LOF with 99.65% [2].

In 2019, an efficient machine learning model, which includes common classification techniques such as ELM, KNN, Decision Trees (DT), MLP and SVM has been proposed. This hybrid model which is a combination of MLP, DT & SVM [10]. Among the all, SVM has achieved the highest accuracy of 81.63%. But the proposed model is more robust and has minimum false alarm rate, with an accuracy of 82.58%, which is higher than any of the individual classifier's accuracy [3].

979-8-3503-4745-6/23/$31.00 ©2023 IEEE

We used a dataset which is highly imbalanced, to overcome this, various sampling techniques are being employed. They used an ensemble method, which is a combination of 3 feed-forward neural networks (L1 with 3 hidden layers, L2 with 2 hidden layers & L3 with 2 hidden layers) and 2 random forest classifiers having 300 and 400 decision trees respectively [8]. To train all the networks, ADAM algorithm has been used, with a learning rate of 0.0005. In their experiment, random forest has worked more accurately in detecting normal instances and neural networks for detecting fraudulent ones [5].

Worked built using fraud detection model which works on decision tree algorithm, logistic regression and KNN algorithm [9]. They have presented the individual performance metric of each algorithm at different fraud rates. Among them, logistic regression has performed well and gave best accuracy and sensitivity results. They also stated that there's no algorithm that is universally good. But a combination of multiple algorithms could give us better results [6].

In research employing with CNN model improvises the categorization accuracy of traffic signs. The trained model, Yolov4-tiny is utilized, which is the fourth iteration of the well-known You Only Look Once model that prioritises speed above accuracy. Hence, Yolo, one of the top real-time object identification models [4, 7].

III. Preliminaries of Proposed Work

A. *Dataset*

The dataset that we have used, was taken from Kaggle. FraudTrain and FraudTest are the datasets used in our experiment, where the former consists of 1296675 records with 23 attributes and the latter contains 555719 records with 23 attributes. It can be understood that both datasets are very highly imbalanced. Therefore, they are balanced with the help of various sampling methods, involving random under-sampling, near-miss sampling, and synthetic minority over-sampling (SMOTE). Due to these sampling techniques, it gets easy for us to deal with a smaller sample of data and also to train the model better. Moreover, there were few categorical variables in the dataset. Therefore, one-hot Encoding, RareLabelEncoder and WoE Encoder were used to fit the model, so that the dataset consists of numerical attributes only. With these steps being performed, the feature engineering has been completed and now our model has got into its correct shape for training.

B. *Nature of Input data*

In the context of any kind of anomaly detection technique, the nature of the input data is a decisive factor. Generally, input data is a set of data instances. Data instances are also known as "objects," "points," "records," "patterns," "vectors," "events," "samples," "observations," and "entities," among other terms. And each data instance has a collection of properties such as "variables, dimensionality, characteristic, feature, and field" that can be used to characterise it. Binary, Categorical, and Continuous characteristics are examples of different types of attributes. The data object is called "Univariate" if it only has one attribute. Else if there are multiple attributes, then it is "Multivariate". Note that while dealing with multivariate data instances, all attributes can have similar type of data or a blend of different data kinds The implementation of anomaly detection techniques is also determined by the nature of the data or attributes. For instance, different statistical models must be employed for categorical or continuous data when using statistical techniques.

C. *Structure of analomoly*

If an individual data instance is found to be abnormal to the rest of the data, i.e., if any particular data instance is found to be deviating from the rest of the data, it is referred to as "Point anomaly" and is presented in Fig.1. An easy way to determine point anomalies is that usually they are in isolation from the other data instances. This is the most fundamental sort of anomaly, and it is the subject of the majority of research.

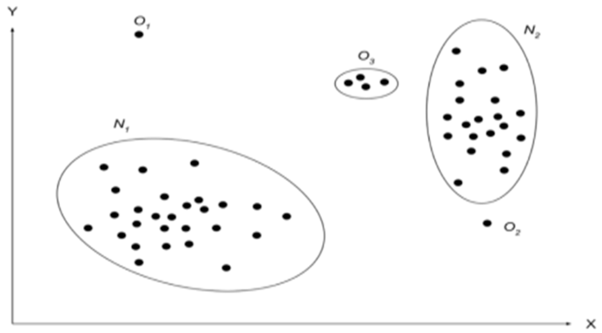

Fig. 1. Point Anomalies

In case, when one data instance in a set is abnormal. condition and not in others, it is referred to as a 'Contextual Anomaly' which is shown in Fig.2. This kind of anomaly works on the basis of specific conditions. Note that contextual anomalies appear as anomalies at different time periods.

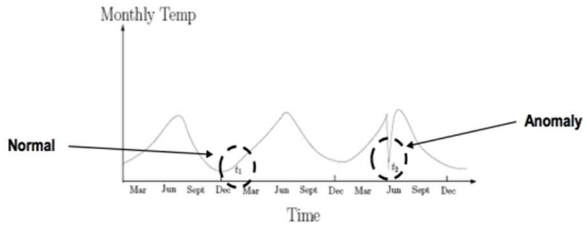

Fig. 2. Contextual Anomaly

A 'Collective Anomaly' occurs when a group of connected data examples does not conform to the entire dataset (presented in Fig.3). Here in case of collective anomalies, the individual data examples that are a part of collective anomaly are not actually anomalous themselves, they are individual data instances are in a group. Else those data instances are individually not anomalous.

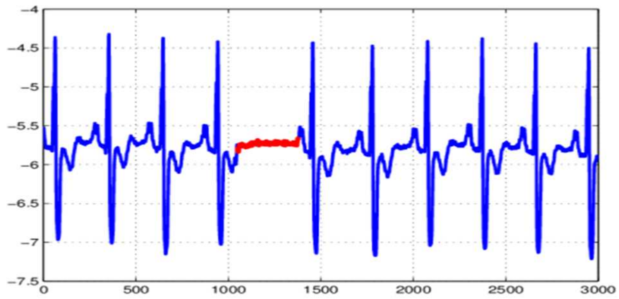

Fig. 3. Collective Anomaly

D. Data labels

Data labels denote if the data instances are normal or anomalous. Manually labelling is a difficult task to do but to obtain labelled training set, a lot of effort is required. Note that the anomalies are dynamic in nature. If there is no labelled training data, chances are that the new anomalies arise. Data labels are of 3 types as presented here.

Supervised anomaly detection means to label each observation, whether they are anomalous or not. Most of the observations are labelled than anomalous. It is similar to that of skewed or imbalanced classification. Any previously unknown data instances are compared to the model to identify which class they fall into if they are found. In the case of supervised anomaly detection, two problems occur. They are: 1. Anomalies are very few compared to the normal data instances. 2. It is really challenging for anomaly class to obtain accurate and representative labels.

Semi-supervised anomaly means to group all the observations have just one label, usually the normal class. It assumes that all data points are similar in the feature space and robust features can be learned to understand if a new point is an anomaly A mix of supervised and unsupervised methods is called semi-supervised learning. This type of technique isn't often employed because obtaining a dataset that covers every possible abnormal behaviour in the data is tough. But semi-supervised algorithms enable the algorithm to re-train on the basis of user feedback upon the anomalies generated. This also helps in rectifying the mistakes made earlier and not to repeat the same mistakes at the later stages. Note that whenever semi-supervised algorithms are integrated, there exist a lot of challenges as well.

Unsupervised anomaly is those which have no labelled data. It depends on the assumptions that it is possible to distinguish normal examples from anomalies geometrically. Because these strategies do not require any training data, they can be used in a wide range of situations. The approach would suffer from a high false alarm rate if the assumptions were incorrect, which would result in the development of more false positives. If the business organizations find higher number of alerts, then they are being prioritized on the basis of scores or they can set a higher threshold value to increase their focus towards critical anomalies. The model developed during training, nevertheless, is resistant to these few anomalies, and test data includes remarkably little of them.

IV. Proposed Methodology

The wide use of online purchases has brought in a lot of challenges and it is often difficult for us to distinguish between a normal and a fraudulent transaction. With an increased growth of credit cards, more are the chances for a fraudster to do irregular transactions. The number of legitimate transactions exceeds the number of illegitimate ones, hence it has difficult for anyone to detect the fraud ones, as they are very less in number. Therefore, it has become a challenging task for researchers to automatically detect the transaction as normal or anomalous. Fig.4 presents the general flow of fraud detection approach.

The study's primary goal is to use various machine learning and deep learning approaches to categorise the dataset's transactions as fraudulent or not fraudulent [11,14]. Our model works by using different algorithms and gives an improved accuracy score by predicting with minimized misclassification. Another important objective of this model is to identify all the fraudulent transactions with a greater accuracy, while neglecting the incorrectly classified transactions. It also aims at maximizing the accuracy of fraud detection while minimizing the false negatives. Moreover, the model is estimated to detect all the fraudulent transactions. Lastly, it shows us which algorithm performed better and one pick the best approach from it [15].

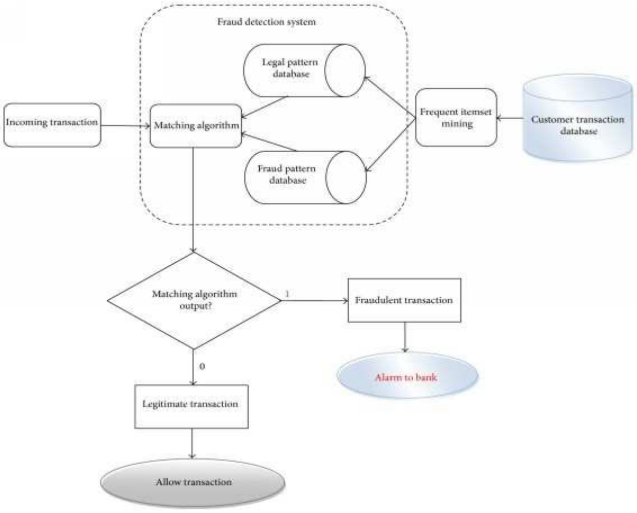

Fig. 4. Fraud detection approach

V. Output of Anaomoly Detection

The output that are reported by any anomaly detection are of two types.

A. Scores

Using scoring methods, each occurrence in the dataset is given an anomaly score depending on how much it is deemed abnormal. Hence the output comes out to be list of ranked anomalies, i.e., based on the scores, a list of anomalies is being given as output. Depending upon the need of an analyst, he can analyse only top few anomalies or selecting a few anomalies by setting up a threshold limit. With the threshold limit, all the anomalies that fall under that limit are selected or can be retrieved as an output.

B. Labels

Each instance is assigned with a label if is normal or anomalous. This is a kind of classification-based approach, where the data is classified into various classes. Here we have binary class i.e., normal and anomalous classes. Therefore, it is easy for us to determine which is normal data and anomalous data.

C. Autoencoders

An artificial neural network variant is the autoencoder. It is an unsupervised learning method that compresses data well and is employed for both encoding and decoding. The function f is used by the autoencoder to encrypt the input values x (presented in Fig.5). Then, it applies a function g to the encoded values f(x) to decode them, producing output values that are analogous to the input values. Let us take an example to explain the working of autoencoders. If the model is given a dataset as an input, then it goes to the first part that is the encoder. Here in this part, the model tries to understand the high-level features automatically. And it also learns how to reduce to lower dimensions. Then the input file is compressed to an encoded representation. This encoded representation is transmitted to the Bottleneck part. The bottleneck layer (also known as Hidden Layer) is the most compressed form of input data and it is the latent space representation of the input. Here it is compressed to the lowest levels and the low-level features are learnt by the model. This way it makes itself familiar to recognize the normal data.

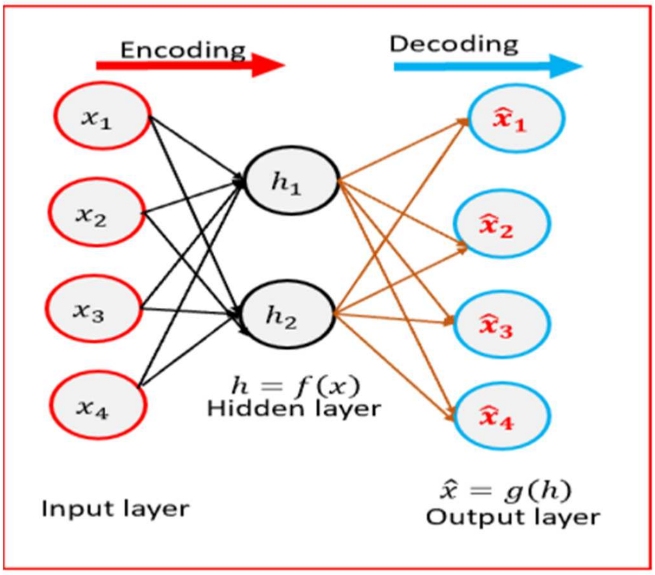

Fig. 5. Auto encoders and its layers

Now the compressed representation is again transmitted to the decoder part. In the decoder part, model learns to reconstruct the data from the encoded representation such that the reconstructed data is similar to the input. We need to note a point here that the output of decoder is very similar to the input of the encoder, except that there's a loss of pixels in the decoded output but the data is same. Later, the decoded output is back-propagated again and again to minimize the reconstruction loss.

Therefore, the output is similar to the input. The model performs well to give best results. At last, the model measures how better the decoder is working and how close is the output to the input. The model calculates the reconstruction loss i.e., the deviation of the output from the input. Thus, formed deviation is therefore used as an anomaly score [12].

Here are the steps for the implementation of Autoencoders:

Step 1: Get the dataset ready
Step 2: Encode the input into a different vector h. The input vector has a higher dimension than h. h = f(Wx+b)
Step 3: Involves decoding the vector h that serves as the input. The output will be the same size as the input.
Step 4: Determine the reconstruction error. Reconstruction Error = input vector - output vector, where To update the weights
Step 5: Back-propagate the mistake from the output layer to the input layer. For each observation in the dataset, repeat steps 1 through 5 in step 6.
Step 7: Go through additional epochs.

D. Isolation Forest

Isolation forest is an unsupervised machine learning approach for anomaly identification that isolates outliers in the data to identify anomalies. It is built upon the Decision Tree algorithm. It distinguishes outliers by randomly choosing a feature from a collection of features, and then randomly choosing a split value between the feature's minimum and maximum values. Only a few random divisions are needed to isolate the anomalies from the dataset's normal points. As a result, then anomaly route will undoubtedly be shorter.

The process of anomaly detection often starts with the creation of a profile of what is "normal," following which anything that deviates from that profile is labeled as "anomalous." On the other hand, the isolation forest method does not work under this assumption and does not calculate point-based distances. The advantages of Isolation Forest are that we detect anomalies very fast, reduce the s computational cost, have linear time complexity, and also demand less memory in comparison to the other algorithms [4].

E. Local Outlier Factor (LOF)

The local density deviation of a data point in relation to its neighbours is determined using the unsupervised anomaly detection technique known as LOF. If the density of a sample differs considerably from that of its neighbours is considered an outlier. The anomaly score is determined for each sample. The LOF score indicates how probable a data point is to be an outlier or abnormality. LOF ≈1 ⇒ no outlier LOF ≫1 ⇒ outlier A local outlier is a site that is deemed an outlier based on its immediate surroundings. LOF will detect an outlier based on the density of the surrounding area. When the data density is not uniform throughout the collection, LOF performs effectively. Note that, Outliers are the points with the largest LOF value [4]. The steps involved in the implementation of our model are summarized as shown below:

Step 1: Importing the libraries and Loading the dataset
Step 2: Performing Exploratory Data Analysis
Step 3: Converting categorical variables (if any) to numerical
Step 4: Handling the imbalanced data with sampling techniques
Step 5: Building an Autoencoder Model with four layers
Step 6: Training the model with Tanh and ReLU activation functions and various optimizers like RMSProp, ADAM, Adadelta
Step 7: Training the model with varying batch sizes and checking the improvement in accuracy
Step 8: Training and Testing the model with Isolation Forest and LOC algorithms
Step 9: Comparing the accuracy of both trained and test datsets
of IF and LOC algorithms

VI. Results and Discussions

The comparison of auto-encoders with different parameters is presented in Table 1. Table 2 and Table 3 presents comparison of auto-encoders with different batch size with Epoch-50 and Epcoh=100 respectively. The experimental analysis, the results of different approaches with different models were noted and presented in Table 4.

The results are evaluated using parametric evaluation as presented in Table 1-4 with Precision, Recall, F1-score and accuracy as evaluated by equations (1) to (4) respectively.

$$Accuracy = \frac{(TN+TP)}{(TN+FP+TP+FN)} \tag{1}$$

$$Precision = \frac{TP}{TP+TF} \tag{2}$$

$$Recall = \frac{TP}{FP+FN} \quad (3)$$

$$F1 - Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

The output “TN” stands for True Negative which shows the number of negative examples classified accurately. Similarly, “TP” stands for True Positive which indicates the number of positive examples classified accurately. The term “FP” shows False Positive value, i.e., the number of actual negative examples classified as positive; and “FN” means a False Negative value which is the number of actual positive examples classified as negative.

TABLE I. COMPARISON OF AUTO-ENCODERS WITH DIFFERENT PARAMETERS

Model	Precision	Recall	F1-Score	Accuracy
Autoencoder (Neurons 200,100,50,100,200; Batch Size=32)	0.02	0.15	0.03	94.6216
Autoencoder (Activation Func= tanh)	0.02	0.16	0.03	94.6278
Autoencoder (Optimizer = RMSProp)	0.02	0.15	0.03	94.6216
Autoencoder (Optimizer = Adadelta)	0.02	0.15	0.03	94.6216

TABLE II. COMPARISON OF AUTO-ENCODERS WITH DIFFERENT BATCH SIZE WITH EPOCH-50

Model	Precision	Recall	F1-Score	Accuracy
Autoencoder (Batch Size=256)	0.02	0.21	0.03	92.6859
Autoencoder (Batch Size=512)	0.02	0.21	0.03	92.6859
Autoencoder (Batch Size=1024)	0.02	0.21	0.03	92.6859

TABLE III. COMPARISON OF AUTO-ENCODERS WITH DIFFERENT BATCH SIZE WITH EPOCH-100

Model	Precision	Recall	F1-Score	Accuracy
Autoencoder (Batch Size=256)	0.02	0.21	0.03	92.6890
Autoencoder (Batch Size=512)	0.02	0.21	0.03	92.6828
Autoencoder (Batch Size=1024)	0.02	0.21	0.03	92.6859

TABLE IV. COMPARSION OF DIFFERENT MODELS AND PERFORMANCES

Model	Precision	Recall	F1-score	Accuracy
LOF	0.5	0.8	0.8	99.0005
IF	0.01	0.01	0.01	98.9264
Autoencoder (Neurons 14,7,7,29; Batch Size=32)	0.03	0.22	0.06	96.4082

VII. CONCLUSIONS AND FUTURE SCOPE

After conducting a comparison analysis, it was discovered that each approach has its own/ strengths and disadvantages. There are several anomaly detection strategies accessible because only a tiny percentage of transactions are fraudulent. None of them are able to identify fraud while it is occurring; instead, they do it after it has already occurred. Therefore, the solution to this problem is to develop technology which can identify fraud as soon as it occurs. Therefore, a key objective of ours is to develop a model that performs well and can identify financial fraud in relation to credit card transactions. This model divided the dataset into training and testing stages after concatenating the two datasets. Implementation of the isolation forest and local outlier factor is then performed, then an autoencoder with four layers and various activation functions is followed by optimizers. From our experimental analysis, it is known that LOF has outperformed IF and Autoencoders with a better accuracy 99% and recall 0.8.

The use of credit cards and internet shopping has increased dramatically in recent years. This widespread demand is extremely risky, as there is a possibility of fraud. As a result, there is a pressing need to halt fraudulent transactions. In this study, machine learning and deep learning methods for identifying fraudulent transactions are examined. However, in the future, we'll use cutting-edge techniques like GANs, Variational Autoencoders, and Sparse Autoencoders to fight fraud. Although difficult and expensive, preventing fraudulent transactions is not impossible. As a result, we assure that we will focus on fraud prevention methods, which ultimately leads to preventing abnormal activity and economic loses to financial institutions.

REFERENCES

[1] M. Zamini and G. Montazer, "Credit Card Fraud Detection using autoencoder based clustering," *2018 9th International Symposium on Telecommunications (IST)*, Tehran, Iran, 2018, pp. 486-491.

[2] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic and A. Anderla, "Credit Card Fraud Detection - Machine Learning methods," *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, 2019, pp. 1-5.

[3] H. Najadat, O. Altiti, A. A. Aqouleh and M. Younes, "Credit Card Fraud Detection Based on Machine and Deep Learning," *2020 11th International Conference on Information and Communication Systems (ICICS)*, 2020, pp. 204-208.

[4] Alam, M., Podder, P., Bharati, S., Mondal, M.R.H. (2021). Effective Machine Learning Approaches for Credit Card Fraud Detection. In: Abraham, A., Sasaki, H., Rios, R., Gandhi, N., Singh, U., Ma, K. (eds) Innovations in Bio-Inspired Computing and Applications. IBICA 2020. Advances in Intelligent Systems and Computing, vol 1372.

[5] D. Prajapati, A. Tripathi, J. Mehta, K. Jhaveri and V. Kelkar, "Credit Card Fraud Detection Using Machine Learning," *2021 International Conference on Advances in Computing, Communication, and Control (ICAC3)*, Mumbai, India, 2021, pp. 1-6.

[6] Y. Pandey, "Deep learning for credit card fraud detection," International Journal of Advanced Research in Computer Science, vol. 8, no. 5, May-June 2017.

[7] C.B. Murthy, M.F Hashmi and A.G. Keskar, “EfficientLiteDet: a real-time pedestrian and vehicle detection algorithm”, Machine Vision and Applications, vol.**33**, Issue. 47, 2022.

[8] I. Sadgali, N. Sael and F. Benabbou, “Fraud detection in credit card transaction using neural networks”, In: Proceedings of the 4th International Conference on Smart City Applications (SCA ’19). Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–4, doi:10.1145/3368756.3369082. Article 95

[9] A. Alshammari, R. Alshammari, M. Altalak, K. Alshammari and A. Alhakamy, "Credit-card Fraud Detection System using Big Data Analytics," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Maldives, Maldives, 2022, pp. 1-7.

[10] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, “Real-time credit card fraud detection using machine learning,” in Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering, vol. 7, no. 10, pp. 488–493, Noida, India, 2019.

[11] Alghofaili, Y., Albattah, A. and Rassam, M. A, ‘A Financial Fraud Detection Model Based on LSTM Deep Learning Technique’, Journal of Applied Security Research, pp. 1–19, 2020.

[12] A. Biswas, R. S. Deol, B. K. Jha, G. Jakka, M. R. Suguna and B. I. Thomson, "Automated Banking Fraud Detection for Identification and Restriction of Unauthorised Access in Financial Sector," *2022 3rd*

International Conference on Smart Electronics and Communication (ICOSEC), 2022, pp. 809-814.

[13] N. C. Matson, D. Rajan and J. Camp, "Design and Analysis of Neural-Network-based, Single-User Codes for Multiuser Channels," *2022 IEEE Latin-American Conference on Communications (LATINCOM)*, 2022, pp. 1-6.

[14] O. Altiti, “Credit card fraud detection based on machine and deep learning,” in 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 204–208, 2020.

[15] X. Yu, X. Li, Y. Dong and R. Zheng, “A Deep Neural Network Algorithm for Detecting Credit Card Fraud”, In 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), pp.181- 183, 2020.