Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766227

A FEDERATED BASED APPROACH FOR FRAUD DETECTION IN TRANSACTION

Naiya Suthar
Research Scholar, Parul University, Vadodara, India
210303205006@paruluniversity.ac.in

Trilok Suthar
Assistant Professor, bParul University, Vadodara, India
trilok.suthar270046@paruluniversity.ac.in

Dhenuka Patel
Assistant Professor, cParul University, Vadodara, India
dhenuka.patel2962@paruluniversity.ac.in

Abstract - Consumers and financial institutions all across the world struggle with financial fraud on a regular basis. Every year, they lose billions of monies. As a result, it's critical to have an efficient Fraud Detection System (FDS) to reduce loss for both customers(users) and financial institutions. Utilizing machine learning algorithms, which aid in future prediction and pattern recognition by analyzing vast amounts of data, is a typical method of detecting fraud. A big dataset is necessary to obtain a well-performing model, yet datasets have the drawback of being skewed. i.e., samples of fraudulent transactions are far less common than samples of honest transactions. Furthermore, banks and other financial organizations often are not permitted to disclose their transaction data due to the data privacy and security connected with transaction datasets. When these issues are combined, it is challenging for the centralized FDS to identify fraud tendencies. In this thesis, present a framework for federated learning, a machine learning environment where numerous entities cooperate to solve a machine learning issue under the supervision of a central server or service provider, to train a fraud detection model. With this strategy, financial institutions may profit from a common model that has witnessed more fraud than each bank individually while avoiding sharing the dataset. As a result, the user's sensitive information is safe guarded. Thesis's findings suggest that when it comes to identifying financial fraud, the federated model (Federated Averaging) may match or even exceed the central model (Multi-Layer Perceptron).
Keywords: Fraud detection, credit card, financial transaction, cashless transaction, fraudulent transaction, machine learning, deep learning

INTRODUCTION

Fraud is a crime where the purpose is to appropriate money. According to The Association of Certified Fraud Examiners [1] (ACFE), fraud is defined as:
"The use of one's occupation for personal enrichment through the deliberate misuse or misapplication of the employing organization's resources or assets."

Because [2] of the availability of digital statistics online today, data are now readily available all over the world. The storage of all information, from small to large, that also has a significant amount, broad range, frequency, and importance for organizations using the cloud. The complete information is available from a huge number of sources, including social media followers, client order patterns, shares, and likes. Financial institutions have conducted and continue to conduct in-depth research to prevent and identify all types of fraud. But fraud is a complicated idea since it encompasses many, ever-evolving behaviors and strategies.

Credit cards [3] streamline offline transactions and relieve users of the burden of waiting for change while using cash. The popularity of credit cards is further encouraged by the rising demand for online purchasing. Many online retailers only take credit cards or similar credit card-based payment options. Credit card fraud rises in tandem with credit card use. Fraudsters employ several techniques to get or purchase credit card information. The victim's account is then utilized to send money or to make purchases directly using this information. To catch the victims off guard, fraudsters frequently quickly use up the available credit on the cards.

Nowadays Because of the development of internet services, users are more drawn to online banking, and this trend has been accelerating recently. Fraud attacks are a major issue when a message is sent via a communication channel. The technology and applications we utilize in our technological world are constantly changing. Fraud detection is a challenging task for many banks and online payment providers. Hackers may now more easily get personal information and perpetrate online fraud using advanced password decoding tools. The centralization of the technique has the drawback that different financial institutes may have witnessed different kinds of fraudulent transactions, which would make it harder for them to spot new kinds of fraudulent transactions. Collaboration amongst financial institutes to discuss any sorts of fraudulent transactions they have come across would be one way to address this. However, since the financial institutes do not want their rivals to know how much or what kind of fraud, they are vulnerable to, such coordination is a delicate topic.

There are many Fraud detection techniques such as Fig 2, Fraud in transactions is detected based on these techniques. Data mining, Neural Network, Machine Learning.

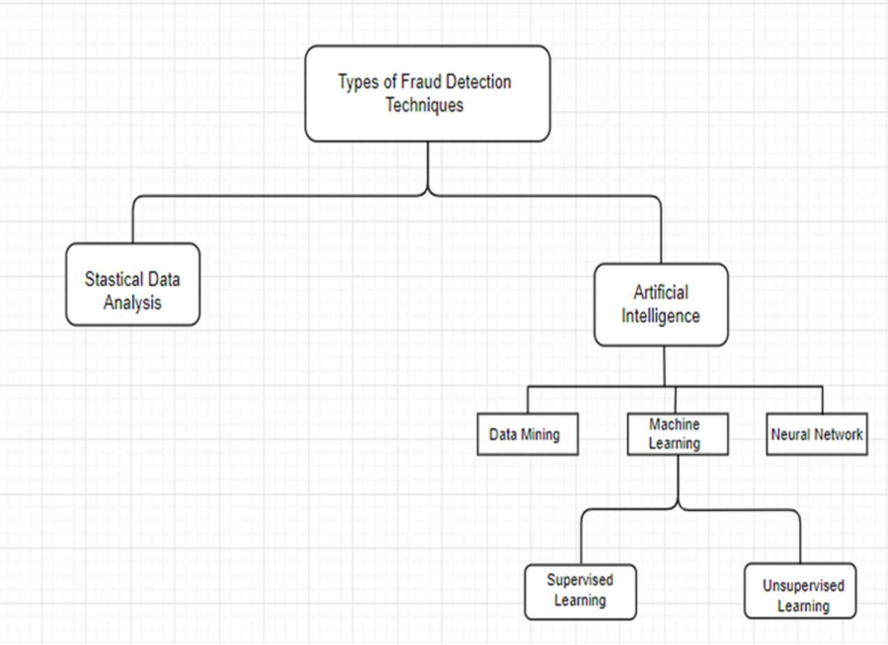

Fig 2: Types of Fraud detection techniques

RELATED WORK

Xiong Kewei., et [6] develop a deep learning-based NN model. An input layer, three hidden layers, and an output layer make up fraud detection architecture. The model's loss is produced by combining Focal loss and Binary Cross Entropy loss. The model may be changed to focus more on the successful records by altering the weights provided to the two classes in the loss using the additional Focal loss parameters, and. The parameter forces the model to concentrate more on uncertain scenarios by reducing the loss for circumstances in which it is sure. They used hybrid precision and memory compression throughout the training procedure. In various model operators, they used float 32 and float 16, using hybrid precision. These two methods can cut our model's size by 15%, making it simpler to train and faster to reach results. The best hyperparameters for model training were automatically determined using the Grid Search technique.

The StackNet model used by Lijie Chen et al. [7] is based on LightGBM, XGBoost, CatBoost, and Random forest. Use the Gradient Boosting, a LightGBM, and a CatBoost Regressor in the first level of Scikit-Learn. The predictions from the level 1 models will be used in level two to train a Random forest Regressor. StackNet controls stacking and cross-validation. A group of lists serves as the model tree's input for StackNet. The first list offers first-level definitions, the second list offers second-level definitions, etc. Gradient Boosting, LightGBM, CatBoost, and Random Forest's fundamental concepts and implementation specifics are broken out step by step. They must thus reveal additional parameters as a result.

Kanika et al. [9] compare 3 thresholding strategies based on the ROC Curve: closest to (0,1) criterion, Youden Index (J), and max-G-Mean in a deep learning-based system for identifying online transaction fraud. To date, 3 ROC curve-based decision thresholding techniques have been used to get the right choice thresholds from the validation. To estimate the likelihood of unclear test results, data will be used. The validation data were used to produce the probabilities

of the DNN model, which were utilized to perform thresholding for each of the 10 folds to find the optimal threshold. Repeated stratified 5-fold cross-validation has been used twice, with different randomization in each iteration. They received a total of 10 folds as a consequence. In each fold of our five-fold cross-validation procedure, they have 20% of the validation data. Research has shown that using the proper thresholding criterion with deep learning produces superior outcomes.

Du Shaohui., et [11] Decision trees are used to create the random forest classifier. Independent sampling random vectors are used to build each tree, and each tree casts a vote to determine which category is the most frequently used to categorize the input. Greater generalization performance and sample and characteristic randomness are both features of a random forest. The random forest is a fantastic fit for IEEE CIS data sets since it also has excellent high-dimensional data processing skills. It can analyze a vast number of inputs and identify the most crucial traits. Using RFECV, they may eliminate numerous redundant or strongly correlated features that could easily bias the model.

Delton Myalil., et [13] conducted studies using both IID and non-IID data. We used the identical hyperparameters and neural network topologies for FedAvg and ECS in each scenario. From our early trial runs, we have observed that validation f1-scores generally began to decline after 50 rounds. Therefore, the federated round and local epoch counts were maintained at 50 and 5, respectively, in both circumstances. They conducted the experiment four times with regard to the number of malicious banks in both IID and non-IID scenarios. First off, none of the cooperating banks were marked as malevolent. Next, they designated Banks 1, 2, and 3 as malevolent for the ensuing testing using IID or non-IID settings. They also trained centralized models on the data for comparison.

K Huang. [18] a fraud detection technique based on LightGBM is suggested in this research. The method takes use of the LightGBM classification model and Bayesian fine-tuning. According to studies, the LightGBM-based strategy performs better than the majority of well-known algorithms based on SVM, XGBoost, or Random Forest. Experiments have been done to evaluate how well the suggested model performs, comparison with machine learning models. The results show that, in terms of AUC and accuracy scores, the model perform better than SVM-based logistic regression, demonstrating it's efficacy in detecting credit card fraud.

Wensi Yang., et [24] Present the FFD detection framework, which uses behavior characteristics and federated learning to train a Federated learning for Fraud Detection model. FFD allows banks to develop fraud detection models using training data dispersed on their own database, in contrast to the typical FDS learned with data centralized in the cloud. Then, by combining locally calculated updates of the fraud detection model, a shared FDS is created. Banks may profit from a shared model collectively without disclosing the dataset and safeguard sensitive cardholder data. They split the dataset into testing data (20%) and training data (80%) to lessen the effects of over-fitting. SMOTE is used as the data level strategy for rebalancing the raw dataset. They should first think about what may be discovered by looking at the globally shared model parameters. Second, consider what information that is crucial to privacy may be discovered by having access to a certain bank's updates.

PROPOSED WORK

The various algorithms are studied. After, analyzing various algorithms in various research papers various authors have implemented different models to identify fraud detection in several types of transactions like Credit card fraud, Online fraud, and UPI payment fraud. To detect fraud in transactions of different financial institutions federated learning model is proposed where various datasets of different financial institutes. It can apply without sharing details with other institutions. Also, reduce the time for training the new model every time. The suggested model is described in fig 3.

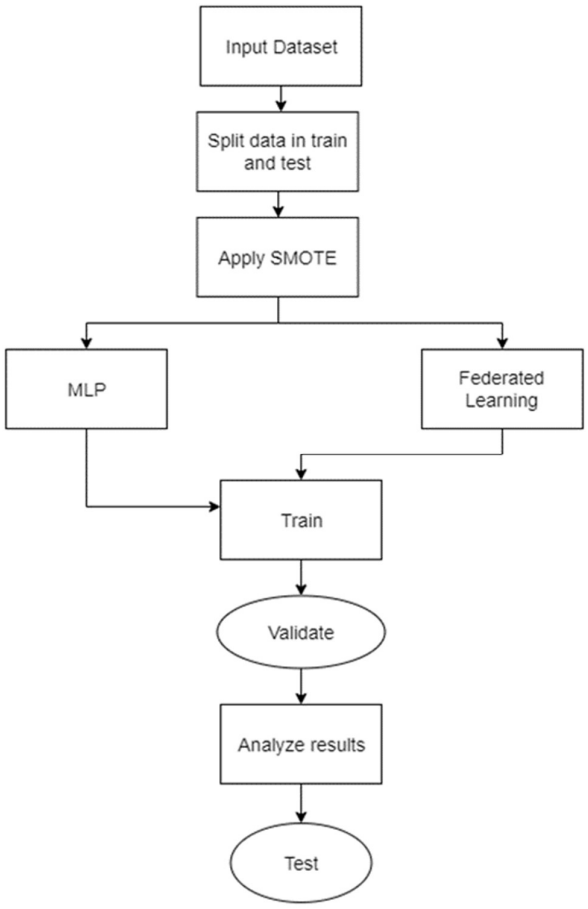

Fig 3: Proposed System

DATA BALANCING

Unbalanced classification is the process of developing prediction models for classification datasets with a large class imbalance. Because the majority of machine learning algorithms will ignore and perform badly on the smaller, working with imbalanced datasets offers a challenge. Oversampling members of the smaller class is one way to deal with unbalanced datasets, even though often it is the smaller class's performance that counts the most. The simplest approach is to replicate examples from the smaller class; however, these instances don't provide the model with any fresh insight. Instead, by combining the previous instances, new ones can be produced. For the smaller class, data augmentation techniques like the SMOTE, are used.

Python's SMOTE for Imbalanced Classification

Creating prediction models for datasets with a considerable class imbalance is known as balanced classification. The challenge with unbalanced datasets is that, despite the fact that performance on smaller class is frequently the most important, most machine learning algorithms will ignore it, leading to subpar results. One strategy for handling unbalanced datasets is to oversample the smaller class. The simplest method is to duplicate instances in the smaller class; however, these examples don't add any new data to the model. Instead, it is possible to synthesize previous instances to produce new ones. For the smaller population, SMOTE, is a data augmentation method. a lack of information from the smaller class, imbalanced categorization makes it difficult for a model to accurately learn the decision boundary. The occurrences in the smaller class can be oversampled as one way to solve this. Before developing a model, this may be achieved by simply reproducing smaller class examples in the training dataset. This could contribute to balancing the class distribution, but it doesn't provide the model any new data. Instead of simply replicating existing examples, it is preferable to synthesis new ones from the smaller class. This type of data augmentation is effective when used with tabular data. Perhaps the most popular technique for creating new samples is the SMOTE.

Nitesh Chawla et al. presented this approach in a 2002 work titled SMOTE: Synthetic Smaller Over-sampling Technique.

SMOTE chooses samples from the spaces with features close to one another, drawing a line connecting the examples, and then drawing a new sample at a location along the line. To be more precise, a random representative from the smaller class is initially picked. Next, for that case, Multi-Layer Perceptron are located. A synthetic example is generated in feature space at a random point between the two cases, using a neighbor that is chosen at random.

Multi Layer Perceptron

Artificial neural networks are a class of algorithms that largely take their cues from how the human brain functions and is organized. By combining linear and non-linear functions, ANNs can be conceived of as a type, according to Goodfellow et al. [28].

$$f = \phi n \circ fn \circ \cdots \circ \phi k \circ fk \circ \phi 1 \circ f1 \quad (1)$$

The creation of parameterized functions f (x, w). In this instance, n is a linear function parameterized by its weights wn, while fn is a nonlinear function. The activation function is commonly referred to as n, and the function fn is known as a layer. The input layer and the output layer are the first and last sets of nodes, respectively, that make up each layer in an ANN. Hidden layers are any groups of nodes that exist between these levels; for further details, the right side of Fig 5. Each weight is often represented as a branch between the layers, and the layers have different attributes depending on how they are linked to the model's nodes. For instance, a basic layer is described as being completely connected if all of its edges are linked to all of its output nodes; as a result, The input layer is combined linearly with the output layer. The single perceptron, shown in Figure 5 with only an input and output layer, is the most basic ANN model. This model computes the output to be a probability between zero and one [29] by utilizing a weighted sum of the input x and a previously established activation function n. This denotes the making of a forward pass.

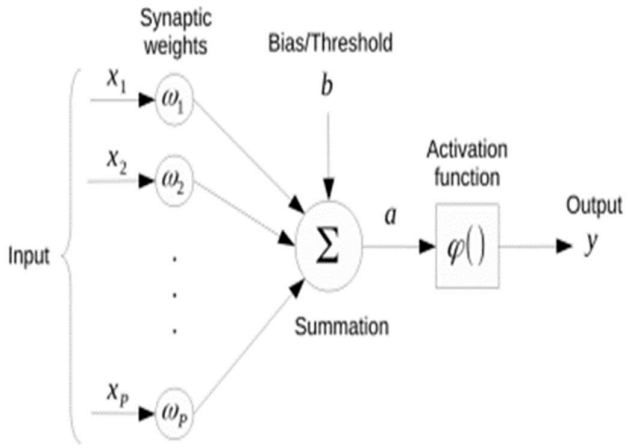

Fig 4: Simple Perceptron with Input and Output Layer

In contrast to the basic perceptron, the MLP comprises having arbitrary numbers of nodes in hidden layers (h), in addition to its p input nodes and m output nodes [30]. According to Goodfellow et al. [28], A feed-forward network, MLP made up of completely linked layers with no recurrent connections. Nowadays, an activation function is applied to each layer, and the Rectifier Activation Function (ReLU) is favored for bigger networks [28]. Moreover, the Logistic activation function, also known as the Sigmoid, is applied to at the end of nodes when working with a binary classification issue, resulting in an output that ranges from zero to one [28]. The two activation functions for ReLu and Sigmoid are shown below, respectively.

$$\phi(x) = \begin{cases} 0 \; if \; x < 0 \\ x \; if \; x \geq 0 \end{cases} \quad \text{..... (2)}$$

and

$$\phi(x) = \frac{1}{1+e^{-x}} \quad \text{..... (3)}$$

where (x) denotes the activation function, which is also displayed on the left side of Fig. 5. Every node in every layer receives the application of this function., as demonstrated by the fact that it appears on every node in Fig 6.

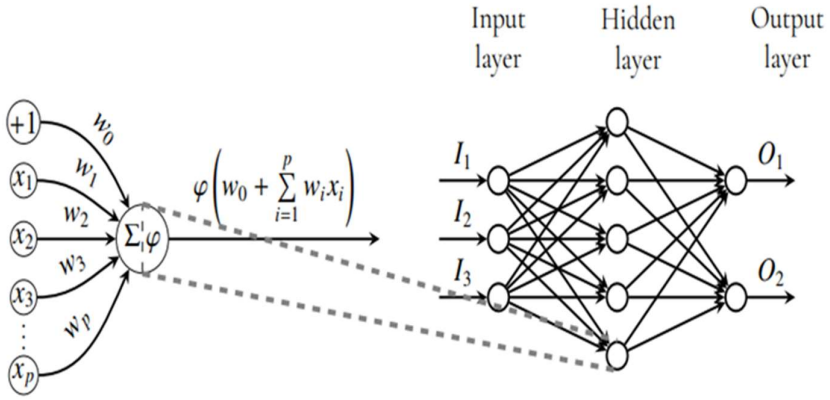

Fig 5: Left: Simple Perceptron, Right: Multi-Layer Perceptron.

In this paper federated learning is used to train and evaluate data in order to identify fraud. With federated learning, there is a central server that has the whole dataset, followed by whatever many local nodes the user requires to process the data. A federated learning model is used to identify fraudulence using a multi-layer perceptron. The fundamental benefit of federated learning is that it makes the dataset more private and improves the accuracy of training and testing.

The dataset [27] has a total of 1048575 transactions. The dataset contains a total of 11 columns of data.

Calculation Parameters

Here, in fig 4 calculation parameters are shown. Accuracy, Recall, Precision, Specificity, and Sensitivity are the classification parameters. Consider these parameters to analyze the result.

	Positive	**Negative**	
Positive	True Positive (TP)	False Negative (FN) Type II Error	**Sensitivity** $\frac{TP}{(TP+FN)}$
Negative	False Positive (FP) Type I Error	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
	Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Fig 6: Comparison Parameters

EXPERIMENTAL RESULT

Three Python libraries Sklearn, NumPy, and Tensor Flow are employed for data analysis, mathematical operations, categorization, prediction, and the creation of data flow graphs. The estimate accuracy is 98%. Precision, Recall and F1- Score are 98%, 90.7% and 31% respectively.

PERFORMANCE PERCENTAGE

Table 1: Performance Percentage

	Decision Tree	**Logistic Regression**	**Random Forest**	**SVM**	**Federated Learning**
Accuracy	97.94	96.94	97.95	98.9	99.3
Precision	97.61	98.59	94.31	92.7	98
Recall	46.06	48.44	67.68	64.1	90.7
F1-Score	62.59	64.96	78.81	64.1	31

Performance Comparison of SVM and Federated learning:

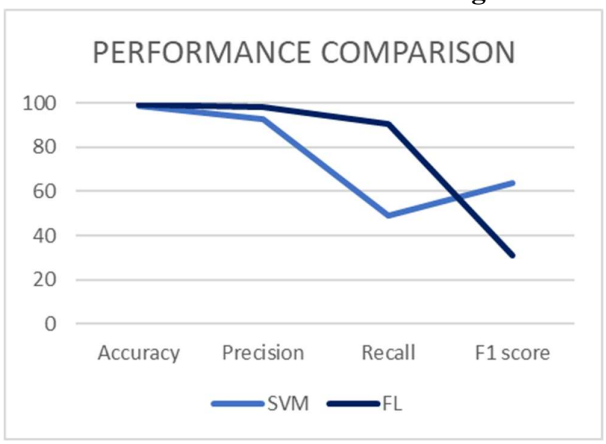

Fig 7.1: SVM, Federated learning

Performance Comparison of Random forest, SVM and Federated learning:

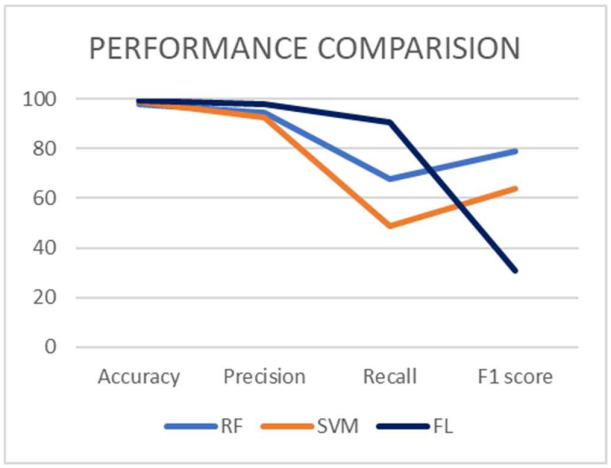

Fig 7.2: Random forest, SVM, Federated learning

Performance Comparison of Logistic regression, Random forest, SVM and Federated learning:

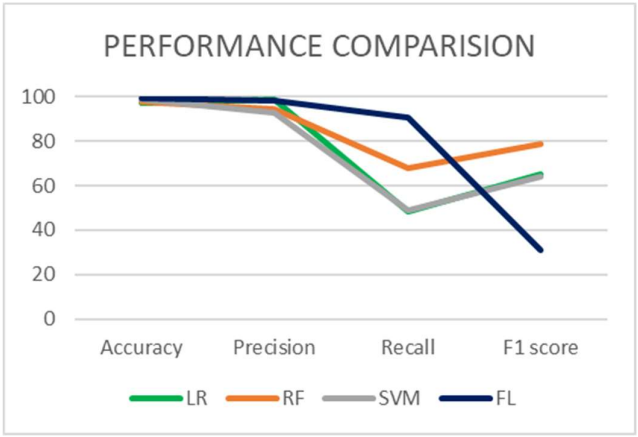

Fig 7.3: Logistic regression, Random forest, SVM, Federated learning

Performance Comparison of Decision Tree, Random forest, SVM, Logistic regression and Federated learning:

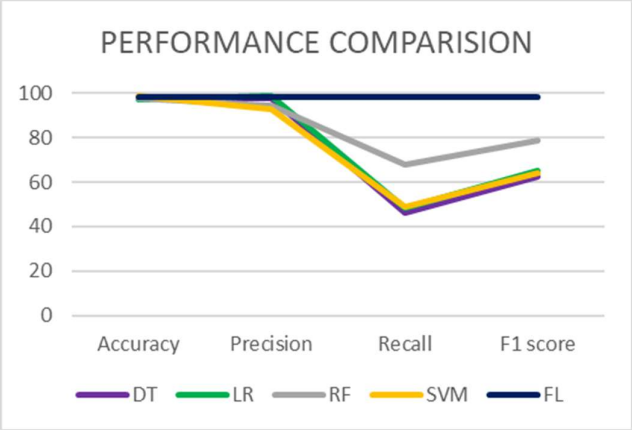

Fig 7.4: Decision tree, Logistic regression, Random forest, SVM, Federated learning

CONCLUSION

Online fraud detection is currently a worldwide epidemic. When fraudsters produce erratic patterns that resemble the original, a more effective method of identifying online frauds while protecting users' privacy is required. In this case, online scams are detected while maintaining privacy using Deep Learning, Machine Learning, and Federated Learning techniques. Comparing the existing techniques with Federated Learning with MLP, where the accuracy has been increasing. SVM is overfitting for large dataset, which is used in this research.

In future work proposed a split learning can be implemented and tested with different machine learning methods.

REFERENCES

[1] The Association of Certified Fraud Examiners, Report to the nations on occupational fraud and abuse, 2014.

[2] Expert System Te@m. What is M@chine Learning? A definition, (2017, March). Url: https://expertsystem.com/machine-learning-definition/. Accessed: 2020-03-06.

[3] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282).

[4] Xia, H., & Ma, H. (2021, April). A Novel Structure-based Feature Extraction Approach for Financial Fraud Detection. In Journal of Physics: Conference Series (Vol. 1865, No. 4, pp. 042101).

[5] Raiter, O. (2021). Applying Supervised Machine Learning Algorithms for Fraud Detection in Anti-Money Laundering. Journal of Modern Issues in Business Research (Vol. 1, No. 1, pp. 14-26).

[6] Kewei, X., Peng, B., Jiang, Y., & Lu, T. (2021, January). A hybrid deep learning model for online fraud detection. In 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (Vol. 9342, No. 110, pp. 431-434).

[7] Chen, L., Guan, Q., Chen, N., & YiHang, Z. (2021, January). A StackNet Based Model for Fraud Detection. In 2021 2nd International Conference on Education, Knowledge and Information Management (Vol. 523, No. 9, pp. 328-331).
[8] Chen, Y., & Han, X. (2021, January). CatBoost for fraud detection in financial transactions. In 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (Vol. 9342, No. 475, pp. 176-179).
[9] Singla, J. (2021, February). Comparing ROC curve based thresholding methods in online transactions fraud detection system using deep learning. In 2021 international conference on computing, communication, and intelligent systems (Vol. 9397, No. 167, pp. 9-12).
[10] Yan, T., Li, Y., & He, J. (2021, June). Comparison of Machine Learning and Neural Network Models on Fraud Detection. In 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (Vol. 9498, No. 212, pp. 978-980).
[11] Shaohui, D., Qiu, G., Mai, H., & Yu, H. (2021, January). Customer Transaction Fraud Detection Using Random Forest. In 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (Vol. 9342, No. 259, pp. 144-147).
[12] Mary, I. M., & Priyadharsini, M. (2021, March). Online Transaction Fraud Detection System. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (Vol. 9404, No. 750 pp. 14-16).
[13] Myalil, D., Rajan, M. A., Apte, M., & Lodha, S. (2021, December). Robust Collaborative Fraudulent Transaction Detection using Federated Learning. In 2021 20th IEEE International Conference on Machine Learning and Applications (Vol. 1, No. 64, pp. 373-378).
[14] Al Smadi, B., AlQahtani, A. A. S., & Alamleh, H. (2021, December). Secure and Fraud Proof Online Payment System for Credit Cards. In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (Vol. 9666, No. 549, pp. 0264-0268).
[15] Wang, L., Zhang, Z., Zhang, X., Zhou, X., Wang, P., & Zheng, Y. (2021). A Deep-forest based approach for detecting fraudulent online transaction. In Advances in Computers (Vol. 120, pp. 1-38).
[16] Trivedi, N. K., Simaiya, S., Lilhore, U. K., & Sharma, S. K. (2020). An efficient credit card fraud detection model based on machine learning methods. International Journal of Advanced Science and Technology (Vol, 29, No. 5, pp. 3414-3424).
[17] Song, Z. (2020, June). A data mining based fraud detection hybrid algorithm in E-bank. In 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (Vol. 978, No. 16, pp. 44-47).
[18] Huang, K. (2020, November). An optimized lightgbm model for fraud detection. In Journal of Physics: Conference Series (Vol. 1651, No. 1, pp. 012111).
[19] Sarma, D., Alam, W., Saha, I., Alam, M. N., Alam, M. J., & Hossain, S. (2020, July). Bank fraud detection using community detection algorithm. In 2020 Second International Conference on Inventive Research in Computing Applications (Vol. 5374, No. 2, pp. 642-646).
[20] Askari, S. M. S., & Hussain, M. A. (2020). IFDTC4. 5: Intuitionistic fuzzy logic based decision tree for E-transactional fraud detection. Journal of Information Security and Applications. (Vol. 52, pp. 102469).

[21] Delecourt, S., & Guo, L. (2019, June). Building a robust mobile payment fraud detection system with adversarial examples. In 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (Vol. 978, No. 26, pp. 103-106).
[22] Singh, A., & Jain, A. (2019). Financial fraud detection using bio-inspired key optimization and machine learning technique. International Journal of Security and Its Applications, (Vol. 13, No. 4 pp. 75-90).
[23] Raghavan, P., & El Gayar, N. (2019, December). Fraud detection using machine learning and deep learning. In 2019 international conference on computational intelligence and knowledge economy (Vol. 987, No. 1, pp. 334-339).
[24] Yang, W., Zhang, Y., Ye, K., Li, L., & Xu, C. Z. (2019, June). Ffd: A federated learning-based method for credit card fraud detection. In International conference on big data (Vol. 11514, pp. 18-32).
[25] Kunlin, Y. (2018, December). A memory-enhanced framework for financial fraud detection. In 2018 17th IEEE International Conference on Machine Learning and Applications (Vol. 978, No. 18, pp. 871-874).
[26] J. Y. Ryu, H. U. Kim, and S. Y. Lee, "Deep learning enables high-quality and high throughput prediction of enzyme commission numbers," Proc. Natl. Acad. Sci. U. S. A., (vol. 116, no. 28, pp. 13996–14001, 2019).
[27] https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset
[28] Goodfellow Ian, Bengio Yoshua and Aaron Courville. Deep Learning. MIT Press, 2016. Url: http://www.deeplearningbook.org. 20
[29] Imane Sadgali, Nawal Sael and Faouzia Benabbou. Lecture Notes on Intelligent Transportation and Infrastructure – Comparative Study Using Neural Networks Techniques for Credit Card Fraud Detection [p.287–296]. Springer, 2019.
[30] Olsson Mattias and Edén Patrik. Lecture notes – Introduction to Artificial Neural Networks and Deep Learning, 2019. Url: https://liveatlund.lu.se/ departments/theoreticalPhysics/FYTN14/FYTN14_2019HT_50_1_NML_1281/CourseDocu ments/Chapt_Intro.pdf. Accessed