Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

An Emerging Technology Of Fraudlent Detection For Enhancing Deep Learning Based Block Chain Innovation

Sowmiyasree.S[1] , Ranganayaki.T[2]

[1]Research Scholar Department of Computer Science,
Erode Arts and Science College, Bharathiar University, TamilNadu, India
[2]Associate Professor, Department of Computer Science, Erode Arts and Science College,
Bharathiar University, TamilNadu, India.

Recently, machine learning (ML) techniques have been highly effective in addressing the problem of payment-related fraud detection. These methods have the ability to evolve and uncover new, previously unseen fraud patterns. In this study, we apply multiple ML techniques, specifically Logistic Regression and Support Vector Machine (SVM), to detect payment fraud using a labeled dataset of transaction data. Our results demonstrate that these approaches can accurately identify fraudulent transactions while maintaining a low rate of false positives. ML, a branch of artificial intelligence (AI), allows systems to learn from data, recognize patterns, and make decisions with minimal human intervention. By leveraging algorithms that build predictive models from training data, ML enables automated decision-making without the need for explicit programming. Deep learning, a more advanced subset of ML, employs deep neural networks with multiple layers, which are particularly effective in processing large, complex datasets.

Keywords: Machine Learning, Deep Blockchain Framework, BiLSTM, Deep Learning Algorithms, Random Forest.

INTRODUCTION

We are witnessing a rapid shift toward digital payment systems, with credit card and payment companies experiencing significant growth in transaction volumes. For example, in the third quarter of 2018, PayPal Inc., a San Jose-based payments company, processed $143 billion in total payment volume [4]. However, this digital transformation has also led to a sharp rise in financial fraud within these systems. To combat this, an effective fraud detection system must accurately and efficiently identify fraudulent transactions. While preventing fraudulent activities is essential, it is equally important to ensure that legitimate users have uninterrupted access to the payment system. A high number of false positives can result in poor customer experiences, potentially driving customers away.

One of the key challenges in applying machine learning (ML) to fraud detection is dealing with highly imbalanced datasets. In most cases, the vast majority of transactions are genuine, with only a small fraction being fraudulent. This imbalance makes it difficult to design a detection system that minimizes false positives while effectively identifying fraudulent transactions. In our paper, we apply several binary classification techniques—Logistic

Regression, Linear SVM, and SVM with RBF kernel—on a labeled dataset of payment transactions. Our objective is to develop classifiers that can distinguish fraudulent transactions from legitimate ones. We also compare the performance of these approaches in terms of their ability to detect fraud.

RELEVANT RESEARCH

Various machine learning (ML) and non-ML-based approaches have been applied to address payment fraud detection. Paper [1] provides a comprehensive review and comparison of several state-of-the-art techniques, datasets, and evaluation criteria used in this domain. It covers both supervised and unsupervised ML approaches, including Artificial Neural Networks (ANN), Support Vector Machines (SVM), Hidden Markov Models (HMM), and clustering methods. Paper [5] introduces a rule-based approach to solving fraud detection problems, while paper [3] highlights the issue of imbalanced datasets, which can lead to a high number of false positives, and presents methods to mitigate this challenge. In [2], the authors propose an SVM-based approach to detect metamorphic malware, addressing the challenge of imbalanced data—specifically, the scarcity of malware samples compared to benign files—and demonstrating how to achieve high precision and accuracy in detection.

LITERARURE SURVEY

Al Shorman, Areej, et al. (2020) discuss how advancements in technologies like artificial intelligence and radio programming are making range-detecting recognition more feasible. However, alongside these advancements come security concerns. To address these, a protection strategy against spectrum sensing data falsification (SSDF) attacks has been proposed, leveraging dual-limit energy detection and blockchain technology [1].

Ali, O., Ally, M., Clutterbuck, and Dwivedi, Y. (2020) debate the balance between prioritizing commercial development and supporting field adaptation. Risks in the data-driven environment highlight various threats, which have been mitigated using BDmarks for watermarking to enhance adaptation and protect against external and internal threats, thus boosting security measures for business growth [2].

Garg, P., Gupta, B., Chauhan, A.K., Sivarajah, U., Gupta, S., and Modgil, S. (2020) focus on preventing data tampering and verification issues, particularly when dealing with unified workers. By avoiding malicious clients, their approach enhances security through distributed storage management. The study also explores the strengths, weaknesses, opportunities, and threats (SWOT) analysis of blockchain technology, emphasizing its security benefits [3].

Kumari, S., and Kumar, R. (2020) highlight how many intermediaries maintain archives to ensure authenticity and integrity. Despite significant investments, these systems often fail to deliver as promised. Blockchain technology provides a solution, simplifying the task of maintaining authenticity and integrity [4].

According to IBM (n.d.), the Internet of Things (IoT) significantly impacts our daily lives. Due to its distributed nature and large scale, central security goals such as confidentiality, integrity, and availability are challenging in IoT. Blockchain's decentralized framework and

cryptographic algorithms can greatly enhance the security of customer data, providing a more reliable foundation for IoT devices [5].

Kumari, S., and Kumar, R. (2020) discuss the ultra-dense network (UDN), one of the most promising developments in the fifth generation (5G) technology, to address network capacity issues. However, securing user equipment (UE) access to UDN, built on autonomous and dynamic access points (APs), remains a challenge. The UDN is considered a decentralized admission network in the 5G landscape [6].

Osmani, Mohamad, et al. (2020) introduce the Relevance Vector Machine (RVM), which evaluates a small set of fixed basis functions from a large dictionary of potential candidates to create efficient classification and regression models. However, the computational complexity of RVM—O(M3) in time and O(M2) in space, where M represents the size of the training set—makes it impractical for handling very large datasets [7].

Qingquan, H. (2021) notes that with the increasing use of digital applications, tools like MapReduce are used for data preprocessing. However, MapReduce's structure is rigid, which can cause inefficiencies during data preparation, such as tilting. While reducing this issue is possible, it often comes at a cost [8].

Li, X., Jiang, P., Chen, T., Luo, X., and Wen, Q. (2020) describe how supply chain management systems help firms by enabling data sharing and analysis. However, data discrepancies between organizations complicate planning algorithms, making it difficult to rely on accurate data [9].

Naheem, M.A. (2019) proposes an alternative encoding strategy based on blockchain technology, where data from activity sequences and corresponding machines are combined in an activity node and linked with other nodes to form an activity list, utilizing C++'s pointer technology [10].

PROPOSED METHODOLOGY

This study proposes a collaborative approach for e-commerce organizations to incrementally develop robust machine learning algorithms while safeguarding business strategies and addressing privacy concerns. The solution leverages real marketplace data to enhance the robustness of the algorithms and utilizes cutting-edge blockchain technology to create a secure platform for training fraud detection models collaboratively. By incorporating smart contracts, the process is fully automated, ensuring that no functionality within the system can be altered by any party. This establishes a foundation of absolute trust, allowing organizations to safely share data and contribute to the development of fraud detection algorithms.

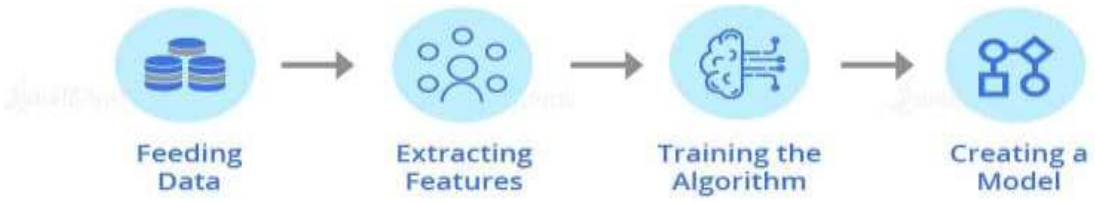


Figure1: Key Stages in the Machine Learning Model Development Process

Feeding Data:

In the first step, data is fed into the model. The performance of the model is largely dependent on the volume of data it is trained on – more data typically leads to a more accurate and robust model. For fraud detection tailored to a specific business, it is crucial to input extensive amounts of data. This ensures the model is trained to accurately identify fraudulent activities specific to that business.

Extracting Features:

Feature extraction involves gathering all relevant information associated with a transaction. This can include details like the location from which the transaction was made, the customer's identity, the payment method, and the network used for the transaction. The following are some of the key features considered:

Identity: This includes checking the customer's email address, phone number, and potentially evaluating their credit score if they are applying for a loan.

Location: The model verifies the IP address and assesses fraud rates based on the customer's shipping address and geographical location.

Mode of Payment: It checks which payment cards were used, the cardholder's name, whether cards from different countries were involved, and the fraud rates of the bank linked to the payment.

Network: The model looks at the number of email addresses and phone numbers used within a particular network for the transaction.

Training the Algorithm: Once the fraud detection algorithm is developed, it must be trained using customer data. This training helps the algorithm learn how to distinguish between fraudulent and legitimate transactions. As the algorithm is exposed to more data, its ability to accurately detect fraudulent transactions improves over time.

Creating a Model:

After training the algorithm on a specific dataset, the model is ready to identify fraudulent and non-fraudulent transactions in a business. One of the significant advantages of using machine learning in fraud detection is that the model continues to improve as it is exposed to larger datasets. Several machine learning techniques are used for fraud detection, and the following example provides insight into one such technique.

Techniques of Machine Learning for Fraud Detection Algorithms:

Fraud Detection Using Logistic Regression:

Logistic regression is a supervised learning technique used when the output decision is categorical, such as ‘fraudulent’ or ‘non-fraudulent.’

Use Case:

Consider a situation where a transaction occurs, and we need to determine whether it is ‘fraudulent’ or ‘non-fraudulent.’ Based on a set of parameters, the algorithm calculates the probability and delivers an output, classifying the transaction as either ‘fraud’ or ‘non-fraud.’

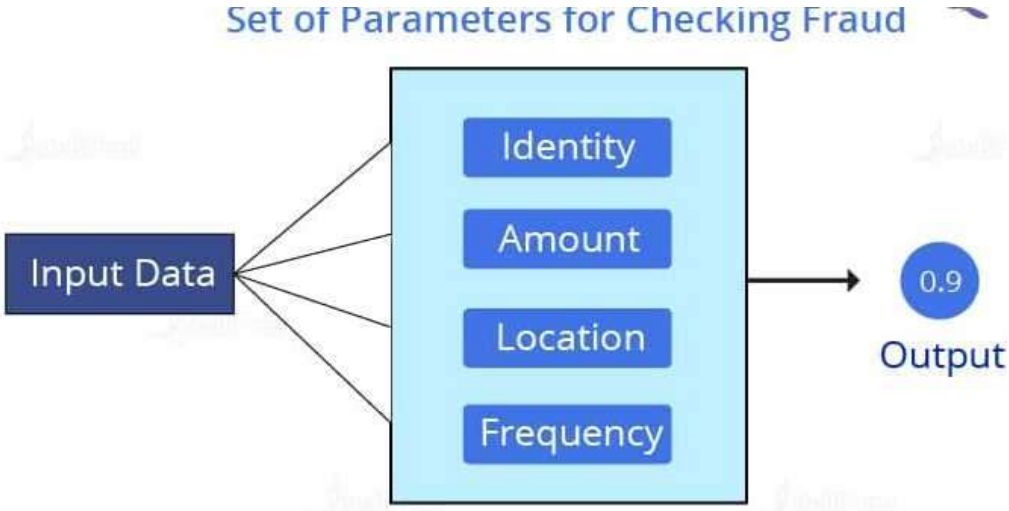


Figure2: Parameters Used for Fraud Detection

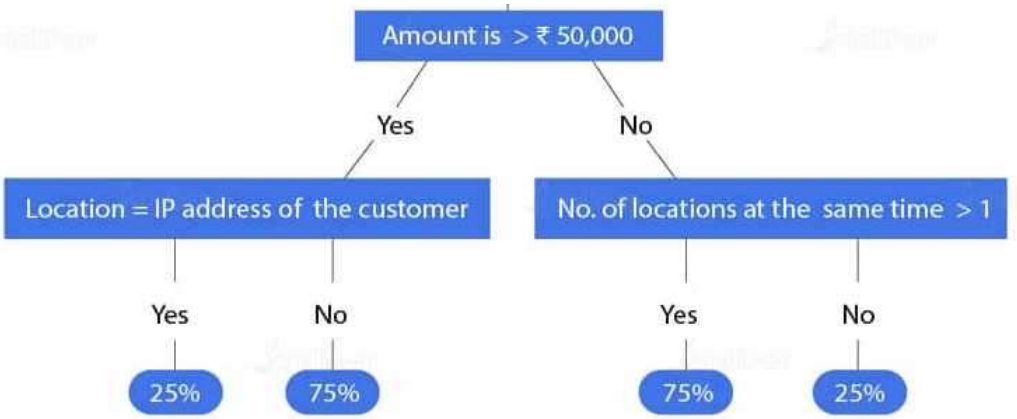
In the diagram above, the calculated probability is 0.9, indicating a 90% likelihood that the transaction is 'genuine' and a 10% chance of it being classified as 'fraudulent.'

Fraud Detection Using Decision Tree Algorithms:

Decision Tree algorithms are particularly useful in fraud detection when there's a need to classify unusual transaction activities from an authorized user. These algorithms utilize a set of constraints and conditions that are trained on a dataset to classify transactions as either fraud or non-fraud.

Use Case:

Consider a scenario where a user initiates a transaction. A decision tree can be built to assess various factors associated with the transaction, allowing the algorithm to predict the probability of fraud based on patterns learned from prior data.



First, in the decision tree, check whether the transaction exceeds 50,000 pounds. In the case of "yes", check where the transaction is executed. And if the answer is no, then we check the frequency of the transaction. After that, we predict the transaction as "fraud" or "non-fraud" according to the probability calculated for these conditions. Here, if the amount is more than ₹50,000 and the location is equal to the customer's IP address, then the chances of "fraud" are 25 percent and "non-fraud" are 75 percent. Similarly, if the amount is more than ₹50,000 and the number of locations is greater than 1, then the chances of "fraud" are 75 percent and "non-fraud" are 25 percent. This is how machine learning decision trees help in creating fraud detection algorithms.

Machine Learning Algorithms for Fraud Detection Using Random Forest: Random forest uses a combination of decision trees to improve results. Each decision tree tests different conditions. They are trained on random datasets, and based on the decision tree training, each tree gives the probability of a transaction being a “fraud” and a “non-fraud”. The model then predicts the outcome accordingly. Use Case: Let us consider a scenario where a transaction is performed. Now let us see how machine learning random forest is used in a fraud detection algorithm.

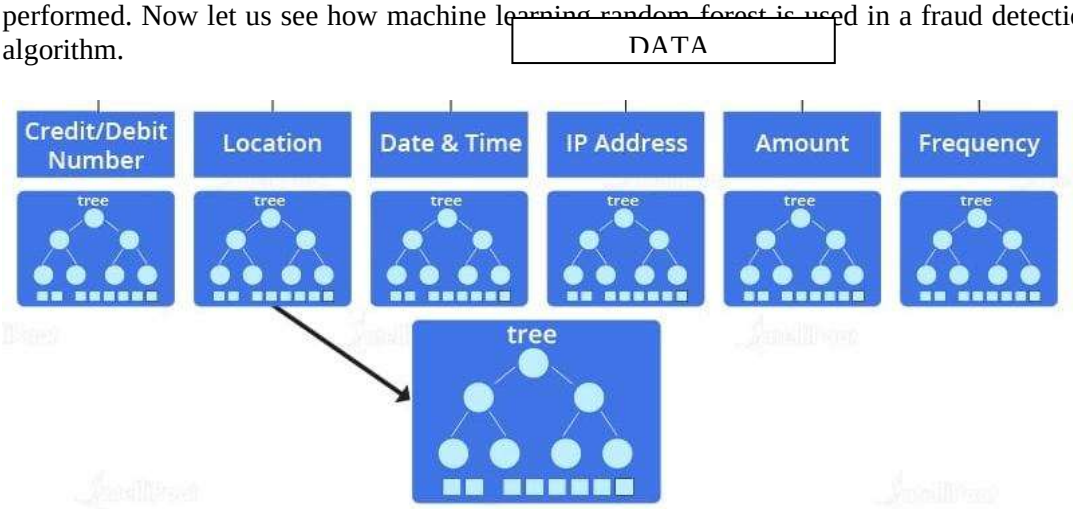


Figure3: Card Processin

When a transaction request is given to the model, it checks information such as credit/debit card number, location, date, time, IP address, amount, frequency of transaction, etc. All this dataset is fed as input to a fraud detection algorithm.[9] The fraud detection algorithm then selects variables from the given dataset that help split the dataset. Therefore, the subsidy consists of variables and conditions, and checks the transactions in which these variables have been approved. After confirming all conditions, all sub -controls may make transactions "fraud" and "non -flaps". '

Algorithm
1. **Initialization**
2: **Inputs:** Minority data M (D)= mi∈ X, Where i = 1, 2, 3, …, D
3: **Outputs:** Synthetic Data S
4: Number of minority samples (D)
5: Percentage of SMOTE (P)
6: Number of (k) nearest neighbors **for** n = 1 to D **do**
8: Find the K nearest neighbors of Di
9: Check P^ =P/100
10: **While** P^ ≠0 **do**
11: Select a random sample m in minority class
12: Find neighbor of m
13: Pick a random number α∈[0,1]
14: $\overline{m}$ =mi+α($\overline{m}$ −mi)
15: **While** Append $\overline{m}$ to S
16: Check $\overline{P}$ =P−1
17: **end while**
18. end

DATASET AND ANALYSIS

In this work, we used the dataset provided by Kaggle [8] simulated mobile payment transactions. We analyze this data by classifying it according to the different types of transactions it contains. We also perform PCA (Principal Component Analysis) to visualize the variability of the data in a two-dimensional space. The dataset contains five transaction categories, labeled "CASH IN", "CASH OUT", "DEBIT", "TRANSFER", and "PAYMENT". The details are shown in Table I [7].

Transaction Type	**Non-fraud transactions**	**Fraud transactions**	**Total**
CASH IN	8786756	0	8786756
CASH OUT	6545343	5643	6545500
TRANSFER	43677	4098	43677
DEBIT	7665644	0	7665644

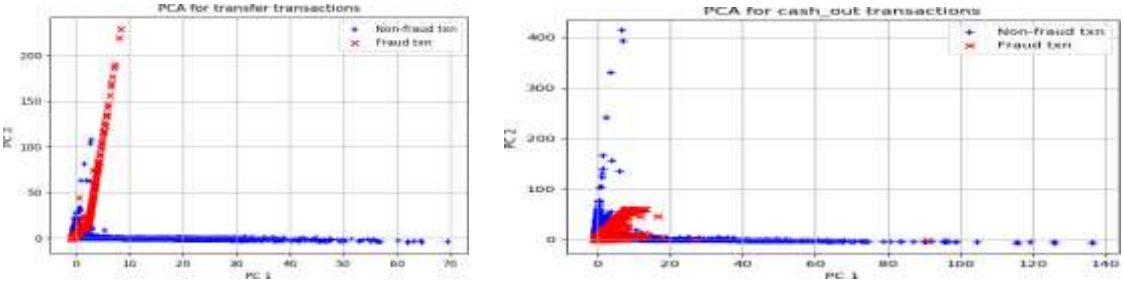
PAYMENT	5678907	0	5678907
TOTAL	3233455	6543	3233500

Table1: Paysim dataset statistics

The Paysim dataset contains numerical and categorical features such as transaction type, amount transferred, sender and receiver account numbers, etc. In our experiments, we train our model using the following features: 1) Type of operation

1. Transaction amount
2. Sender's account balance before the transaction
3. Sender's account balance after the transaction
4. Receiver's account balance before the transaction
5. Receiver's account balance after the transaction

The dataset contains approximately 6 million transactions, of which 8,312 are flagged as fraudulent. This is highly unbalanced, with 0.13% being fraudulent. We present the results of running a 2-dimensional PCA on a subset of two types of fraudulent transactions: TRANSFER and CASH OUT transactions. [8] The PCA decomposition of TRANSFER transactions shows a large variance in the two main components: non-fraudulent and fraudulent transactions. This gives us confidence that the TRANSFER dataset is linearly separable and that our chosen algorithms (logistic regression and linear SVM) are likely to perform very well on such a dataset.



(a) TRANSFER transactions

Figure3: PCA decomposition of Paysim data

METHODOLOGY

Our goal is to divide unauthorized transactions by acquiring a solution boundary in an object space determined by input transactions. Each transaction can be represented as a vector of the sign. Using logistics regression, linear SVM, and SVMs using RBF nuclei, each cash was transferred and built a binary classification device.

A. Logistic regression

Logistic regression is a method used to search for linear solutions of binary classification. Given an input feature vector x, a logistic regression model with parameters θ classifies the input data x using the following assumptions hθ(x) = , which can be interpreted as the probability that x belongs to class 1 as a logistic loss function with respect to parameters θ.

$$K(x, z) = \exp\text{-}\frac{\|x - z\|^2}{2\sigma^2}$$

B. Support Vector Machines

Support Vector Machines create a classification hyperplane in the space defined by the input feature vectors. The goal of the training process is to determine a hyperplane that maximizes the geometric bounds with respect to the labeled input. In this project, we use two SVM variants: Linear SVM and RBF kernel based SVM. RBF kernel-based SVM can find a nonlinear decision boundary in the input space.

C. Class Weighting Approaches

For each of the three techniques, we assign different weights to samples that belong to fraudulent and non-fraudulent classes. Such an approach was used to counter the data imbalance problem, with only 0.13% of fraudulent transactions available to us. In a payment fraud detection system, it is more important to detect potential fraudulent transactions than to ensure that all non-fraudulent transactions are executed without problems. The proposed approach punishes errors performed by incorrect classification of fraudulent samples that accidentally classify non -resistant samples. Compared to non -resistant samples, we trained models (each method) using high -class classes for fraudulent samples.

EXPERIMENTS

In this section, we describe our dataset partitioning strategy and the training, validation, and testing processes we implemented. All software was developed using the Scikit-learn [7][8]ML library. A. Dataset Partitioning Strategy

We divided our dataset based on the different transaction types described in the Dataset section. We use TRANSFER and CASH OUT transactions in particular for our experiments because they contain fraudulent transactions. For both types, we split the respective datasets into three parts: 70% for training, 15% for CV, and 15% for testing. We use stratified sampling to create the training/CV/test parts. Stratified sampling allows us to keep the proportion of each class in the distribution,same as in the original dataset.,The details of the departments are shown in Table II.

Table II: Dataset split details

TRANSFER			
Split	**Fraud**	**Non fraud**	**Total**

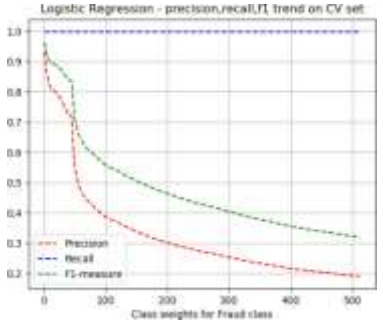
Train	**2868**	**370168**	**373036**
CV	**614**	**79322**	**79936**
Test	**615**	**79322**	**79937**
Total	**4097**	**528812**	**532909**

A. Model training and tuning

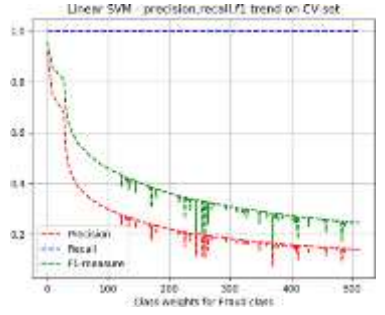
We employed class weight based approach as described in previous section to train our each of our models. Each model was trained multiple times using increasing weights for fraud class samples. At the end of each iteration, we evaluated our trained models, measuring their performance when splitting CV. For each model, we have chosen classes of classes that gave us the maximum review about the fraud class with no more than 1 percent of false works. Finally, we used the model trained with the selected set of class weights to make predictions on the test dataset. In the next section, we discuss in detail the selection of class weights based on their performance on the CV set. We also discuss their performance on train and test sets.

RESULTS AND DISCUSSION

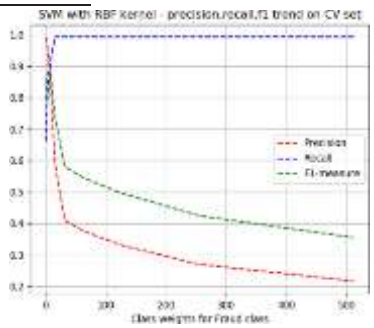
This section describes the results obtained at the stage of training, verification, and testing. The performance of the model was highly evaluated by calculating the measured values such as reviews, accuracy, F1 score, and REZIC curves (auPRC). A. Class Weight Selection In our experiments, we increased the weights for fraud samples. Initially, we considered making the class weights equal to the imbalance ratio in our dataset. This approach seemed to give good recall, but also resulted in a very high number of false positives - >> 1 percent - especially for CASH OUT. Therefore, we did not use this approach and instead tried several combinations of weights in split CV to tune the model. Overall, we found that higher class weights reduced precision but improved recall in split CV. Figure 2 illustrates this behavior as observed for a CASH OUT transaction.[11][12]



a) Logistic Regression



(b) Linear SVM



c) SVM with RBF Kernel

Figure4: : TRANSFER - Precision,Recall, F1 trend for increas- ing fraud class weights

For the TRANSFER dataset, especially for the logistic regression and linear SVM algorithms, the impact of increasing weights is not very noticeable, i.e., equal class weights for fraudulent and non-fraudulent samples result in high recall and precision scores. Based on these results, we always chose a higher weight for fraudulent samples to avoid overfitting in the CV set.When dealing with highly imbalanced datasets, we decided to plot the precision/recall curve (PRC) over the ROC, since PRC is more susceptible to classification errors.

A. Results of trains and test sets

This section describes the results of the train and test set using the weight of the selected class. Figure 5 and Table VI summarize the results of all trains. Get a very high opinion and AUPRC evaluation of transaction transactions with 0.99 test indicators of three algorithms [11]. In particular, SVM with RBF nuclei offers the highest value AUPRC because it has a much higher accuracy than the other two algorithms. Table VIII shows the corresponding confusion matrices obtained on the TRANSFER test set.,All three algorithms are able to detect over 600 fraudulent,transactions with less than 1% false positives. When conducting a principal component analysis of TRANSFER transactions, high variability was found in two principal components.

CONCLUSION

When detecting fraud, we often deal with highly imbalanced datasets. On the selected dataset (Paysim), we show that our proposed approach is able to detect fraudulent transactions with very high accuracy and low false positive rate, especially for TRANSFER transactions [12]. Fraud detection often involves a trade-off between correctly detecting fraudulent samples and not misclassifying many non-fraudulent samples. Our current approach considers the entire set of transactions to train our models. We can create personalized models based on the user's previous trading behavior and use them to further improve the decision-making process. In our opinion, all this could be very effective in improving the classification quality for this dataset.

REFERENCE

1. F. Beena, I. Mearaj, V. K. Shukla, and S. Anwar, ‘‘Mitigating financial fraud using data science—‘A case study on credit card frauds,’’’ in Proc. Int. Conf. Innov. Practices Technol. Manage. (ICIPTM), Noida, India, Feb. 2021
2. S. A. Assefa, D. Dervovic, M. Mahfouz, R. E. Tillman, P. Reddy, and M. Veloso, ‘‘Generating synthetic data in finance: Opportunities, chal- lenges and pitfalls,’’ in Proc. 1st ACM Int. Conf. AI Finance, Oct. 2020, no. 44, pp. 1–8.
3. T. Amarasinghe, A. Aponso, and N. Krishnarajah, ‘‘Critical analysis of machine learning based approaches for fraud detection in financial trans- actions,’’ in Proc. Int. Conf. Mach. Learn. Technol. (ICMLT), May 2018, pp. 12–17.
4. C. Rikap and B. Lundvall, ‘‘Tech giants and artificial intelligence as a tech- nological innovation system,’’ in The Digital Innovation Race. Springer, 2021, pp. 65–90, doi: 10.1007/978-3-030-89443-6_4.
5. J. Lu, W. Li, Q. Wang, and Y. Zhang, ‘‘Research on data quality con- trol of crowdsourcing annotation: A survey,’’ in Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Com- put., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Techno. Congr. DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, Aug. 2020, pp. 201–208.
6. A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective - Samaneh Sorournejad, Zojah, Atani et.al - November 2016
7. Support Vector machines and malware detection - T.Singh,F.Di Troia, Vissagio , Mark Stamp San Jose State University - October 2015
8. Solving the False positives problem in fraud prediction using auto- mated feature engineering - Wedge, Canter, Rubio et.al - October 2017
9. PayPal Inc. Quarterly results https://www.paypal.com/stories/us/paypal- reports-third-quarter-2018-results
10. A Model for Rule Based Fraud Detection in Telecommunications - Rajani, Padmavathamma - IJERT - 2012
11. HTTP Attack detection using n gram analysis - A. Oza, R.Low, M.Stamp - Computers and Security Journal - September 2014
12. Scikit learn - machine learning library http://scikit-learn.org Paysim - Synthetic Financial Datasets For Fraud Detection https://www.kaggle.com/ntnu-testimon/paysim1