International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST52310213

Fraud Transaction Detection Approach Using Machine Learning Hybrid Techniques

B. Ravinder Reddy[1] , N. Rajesh[2], K. V. Anand[2], G. Srikanth[2]

[1]Assistant Professor, Department of CSE, Anurag University, Hyderabad, Telangana, India
[2] Department of CSE, Anurag University, Hyderabad, Telangana, India

ARTICLEINFO

Article History:

Accepted: 20 Feb 2023
Published: 07 March 2023

Publication Issue
Volume 10, Issue 2
March-April-2023

Page Number
90-96

ABSTRACT

Fraud detection is a crucial task in financial transactions to prevent monetary losses and maintain the integrity of the financial system. Some of the machine learning algorithms that we tested for their efficacy in identifying fraudulent activity include Linear Support Vector Classifier (Linear SVC), Support Vector Classifier with Radial Basis Function Kernel (SVC with RBF Kernel), Logistic Regression, Random Forest, Decision Tree, Naive Bayes, Stacking Classifier (Random Forest + SVM with Logistic Regression), and Voting Classifier (Random Forest + Perceptron Algorithm + Boosting. The performance of these algorithms is evaluated using a publicly available dataset consisting of mobile money transactions. We compare the accuracy, precision, recall, F1-score. Our suggested methods demonstrate the capability to accurately identify fraudulent transactions while keeping the number of false positives reasonably low.

Keywords : Fraud detection algorithms include Linear SVC, SVC with RBF Kernel, Logistic Regression, Random Forest, Decision Tree, Naive Bayes, Stacking Classifier, and Voting Classifier.

I. INTRODUCTION

As the world rapidly adopts digital payment systems, the volume of transactions processed by credit card and payment companies is experiencing significant growth.The rapid growth of payment systems has also led to a surge in financial fraud occurring within these systems.

To really battle fake action, an fraud detection framework should have the option to precisely and effectively recognize deceitful exchanges while guaranteeing veritable clients are not kept from getting to the installment framework. The test lies in planning a framework that is coming up short on misleading up-sides while successfully distinguishing fake action, particularly notwithstanding profoundly imbalanced datasets where most of exchanges are real and just a little rate are deceitful. This paper employs several binary classification techniques, including Linear SVM, Logistic Regression, SVM with RBF Kernel, Random Forest, Decision Tree, Naive Bayes, Stacking Classifier, and Voting Classifier. These methods are applied to a labeled dataset containing payment transactions. The objective is to construct binary classifiers that can differentiate between fraudulent transactions and non-fraudulent transactions. Additionally, we seek to evaluate and compare the efficiency of these methods in detecting fraudulent activity.

II. LITERATURE REVIEW

By **Sorournejad et al.** [1], gives an extensive outline of credit card fraud detection procedures from the two information and method situated points of view. The paper orders the strategies in light of their methodologies, including rule-based, information mining, data mining, machine learning, and hybrid approaches. It also provides an overview of different datasets used in the field, along with their strengths and weaknesses. **Singh et al. [2],** investigate the utilization of Support Vector Machines (SVMs) in identifying malware. The authors provide a comprehensive overview of SVMs and their applications in machine learning, including fraud detection. They also explain how SVMs can be used to classify malware as either benign or malicious and discuss the importance of feature selection in the detection process.By **Wedge et al.** [**3**], presents an approach to fraud prediction that focuses on automated feature engineering to reduce the number of false positives. The authors propose an algorithm that utilizes clustering techniques to identify groups of transactions that are similar in nature and then uses these groups to generate features. The approach is evaluated on a real-world dataset, and the results show a significant reduction in false positives. By **Wedge et al.** [**5**], presents an approach to fraud prediction that focuses on automated feature engineering to reduce the number of false positives. The authors propose an algorithm that utilizes clustering techniques to identify groups of transactions that are similar in nature and then uses these groups to generate features. The approach is evaluated on a real-world dataset, and the results show a significant reduction in false positives. By **Oza et al.** [**6**],investigates the utilization of n-gram examination for HTTP assault recognition. The creators use n-gram examination to extricate highlights from HTTP traffic and afterward utilize these elements to prepare a classifier to identify assaults. While the paper focuses on HTTP attack detection, the approach can be applied to fraud detection as well.

III. DATASET AND ANALYSIS

In this study, we utilized a mobile-based payment transactions dataset provided by Kaggle [4]. We grouped the various types of exchanges in the dataset into five categories: Payment, "Cash In," "Cash Out," "Debit," "Transfer," and "Cash In." The Paysim dataset is made out of both mathematical and downright highlights, including exchange type, sum moved, and shipper and beneficiary record numbers. In our trials, we zeroed in on utilizing the accompanying elements to prepare our models: exchange type, exchange sum, shipper account balance before exchange, source account balance after exchange, beneficiary record balance before exchange, and beneficiary record balance after exchange.

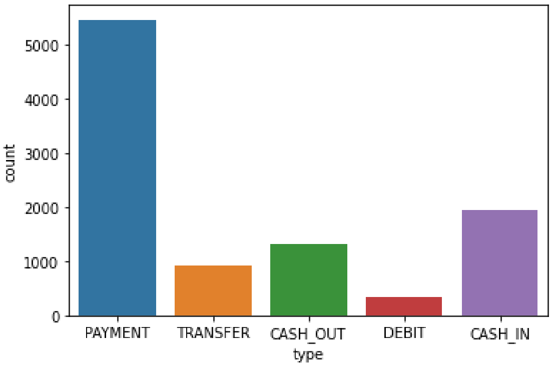


Fig1: Count VS Type of transaction of dataset

We are considering 5465 transactions in payment, 1949 transactions in cash in, 1321 transactions in transfer and 344 transactions in debit.

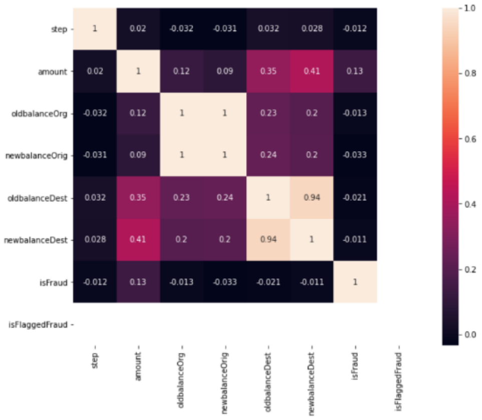


Fig.2: Correlation Matrix

IV. METHODOLOGY

LINEAR SVC

Linear Support Vector Machine (SVM) is a strong characterization calculation that plans to find a hyperplane that can best separate two classes of pieces of information. The hyperplane is a direct capability of the information highlights due to easily distinguishable information. The formula for this is f(x) = wT x + b, where x is the information vector, w is the weight vector, b is the predisposition term, and T is render. The anticipated class mark is determined by the capability yield indicator f(x). The expected class is +1 if f(x) is greater than or equal to 0, and - 1 if f(x) is less than or equal to 0. The weight vector w and inclination term b are mastered during preparing by tackling an enhancement issue that expands the edge while limiting the arrangement mistake. The adequacy of the direct SVM model relies upon the decision of hyperparameters, for example, the regularization boundary C and the decision of bit capability, if any.

SVM WITH RBF KERNEL

The well-known and powerful ML calculation known as the Support Vector Machine (SVM) with the Radial Basis Function (RBF) Kernel is used to investigate order and relapse. The RBF bit is used to move the information into a higher-layered space where a hyperplane can clearly separate the classes. K(x, x') = exp(- gamma ||x - x'||2) is the definition of RBF, where x and x' are input vectors, ||.|| denotes the Euclidean distance, and gamma is a hyperparameter that regulates the part's width. f(x) = sign(sum_i alpha_i y_i K(x_i, x) + b), where alpha_i and y_i are the Lagrange multipliers and class names separately and b is the inclination term, is the choice capability for SVM with RBF portion. The hyperparameters of the SVM with RBF Kernel, including C and gamma, are gotten the hang of during preparing utilizing procedures like cross-approval. SVM with RBF kernel is especially helpful in situations where the information isn't straightly distinct, as it can catch complex non-direct connections between the information highlights and the class names. Be that as it may, the viability of the model relies upon the suitable decision of hyperparameters.

LINEAR REGRESSION

LR models the likelihood of an information test having a place with a specific class, given its feedback highlights. The LR model proposes a calculated capability, otherwise called the sigmoid capability, to plan the information elements to the result likelihood. g(z) = 1/(1 + exp(- z)), where z = wT x + b is the linear function of the information highlights, x is the information vector, w is the weight vector, and b is the predisposition term, is the definition of the strategic capability. Based on the result likelihood, the LR model predicts the class name, using a limit of 0.5 to typically distinguish between the two classes. The LR model parameters, including w and b, are learned during training by maximizing the likelihood of the training data using techniques such as maximum likelihood estimation.

DECISION TREE

Decision Tree (DT) is a strong ML calculation utilized for order and relapse investigation. DT models the dynamic cycle as a tree-like construction, where each interior hub addresses a choice in view of a specific element and each leaf hub addresses a class mark or a mathematical worth. DT algorithm aims to find the optimal splitting criterion that maximizes the information gain or minimizes the impurity measure at each decision node. The impurity measures commonly used in DT algorithm include entropy, Gini index, and classification error. The splitting criterion is determined by comparing the impurity measures of different features and selecting the feature that results in the highest information gain. The DT model is trained by recursively splitting the data into smaller subsets until the leaf nodes are pure or the maximum depth of the tree is reached.

RANDOM FOREST

Random Forest (RF) is a group learning technique that joins numerous choice trees to work on the exactness and security of the forecast. RF calculation makes a bunch of choice trees, each prepared on an irregular subset of the elements and a bootstrap test of the preparation information. The last expectation is gotten by amassing the results of all choice trees. The RF calculation means to lessen the change of the singular choice trees by presenting arbitrariness in the component and information testing. The RF model can handle high-dimensional and noisy data and is less prone to overfitting than a single decision tree. The prediction of RF model is obtained by taking the majority vote or averaging the outputs of all decision trees. The RF algorithm can also provide measures of feature importance based on the reduction in impurity or information gain achieved by each feature. y = sum_i 1(T_i(x) = y)/N is the numerical formula for the expectation of the RF model, where T_i(x) is the output of the I-th choice tree for input x and N is the total number of decision trees in the forest.

VOTING CLASSIFIER

Voting Classifier combines the predictions of multiple base models using a simple majority vote or weighted vote scheme. The base models can be of different types or trained on different subsets of the data. Voting Classifier can handle both classification and regression tasks and can provide better results than a single model, especially when the base models have diverse strengths and weaknesses. The mathematical condition for the assumption for a majority rule classifier can be created as y = argmax_k sum_i w_i * 1(y_i = k), where y is the expected class mark, k is the amount of classes, w_i is the weight of the I-th base model, and 1(y_i = k) is the pointer capacity that benefits 1 if y_i = k and 0 regardless.

STACKING CLASSIFIER

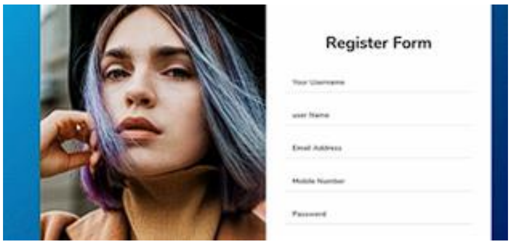
Stacking Classifier prepares different base models on the preparation information and utilizations the results of the base models as info highlights to prepare a meta-model. The meta-model figures out how to consolidate the results of the base models to make the last forecast. Stacking Classifier can deal with both order and relapse errands and can give improved results than a solitary model, particularly when the base models have different qualities and shortcomings. y = f(w_1T_1(x) + w_2T_2(x) +... + w_n*T_n(x)), where y is the expected class mark or relapse value, f() is the meta-model's activation capability, T_i(x) is the result of the I-th base model for input x, w_i is the weight or coefficient of the I-th base model, and n is the number The loads or coefficients can be gained from the preparation information utilizing procedures, for example, linear regression or gradient boosting.

NAIVE BAYES

Naive Bayes is a straightforward probabilistic classifier based on the hypothesis of Bayes and the presumption of restricted freedom between the highlights in the class. Naive Bayes predicts the class name of another event based on the highest deduced likelihood, modeling the likelihood dispersion of the information highlights and the result class. Naive Bayes works well with high-layered and inadequate information and can handle parallel and multi-class characterization tasks. A Naive Bayes classifier's expectation can be expressed numerically as y = argmax_k P(y=k) * prod_i P(x_i | y=k), where y is the expected class mark, k is the number of classes, x_i is the value of the I-th element, and P() is the probability of an event. The class earlier probabilities P(y=k) can be assessed from the preparation information utilizing greatest probability or Bayesian strategies, and the contingent element probabilities P(x_i | y=k) can be assessed utilizing various procedures like most extreme probability, Bayesian assessment, or smoothing.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

Fig 3: Registration of Users


Register Form

Your Username

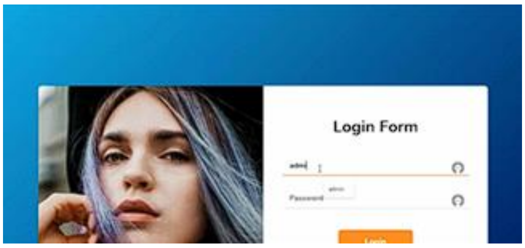
user Name

Email Address

Mobile Number

Password

Fig 4: User Login


Login Form

admin

Password

Fig 5: Page Header

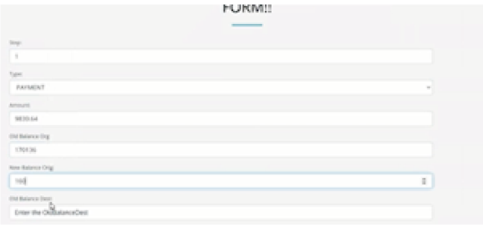
FORM!!

Step:

Enter the Step

Type:

Fig 6: User Input


FORM!!

Step:

1

Type:

PAYMENT

Amount:

9839.64

Old Balance Org:

170136

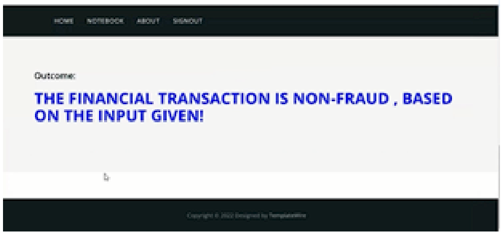
New Balance Orig:

160

Old Balance Dest:

Enter the OldbalanceDest

Fig 7: Result of Prediction



Outcome:

THE FINANCIAL TRANSACTION IS NON-FRAUD , BASED ON THE INPUT GIVEN!

Copyright © 2022 Designed by TemplateMo

Fig 8: Output

Performance Metrics:

In the evaluation process of our models, we utilized an essential step called model evaluation. The performance of the selected models on the test dataset was evaluated in this step using various confusion matrix-based performance measures. For the two classes, True (0) and False (1), the confusion matrix presented the model's arrangement execution on the test set and included classifications such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). To summarize, we evaluated our models' performance on the test dataset by analyzing their classification results using a confusion matrix that includes TP, TN, FP, and FN for the True and Fake classes.

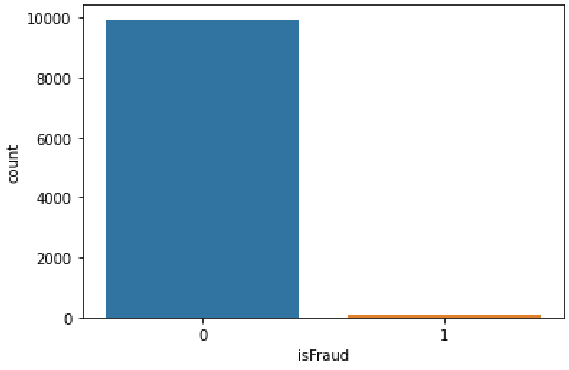


Fig 9: Number of Fraud and non-fraud transactions in the dataset

Accuracy : Accuracy is a widely used metric to evaluate a model's performance. It indicates the percentage of correctly classified instances among all the test cases.

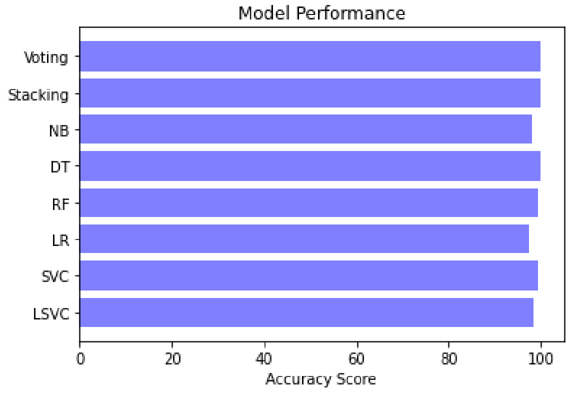


Fig.10: Accuracy

As an extension, we will use Decision Tree, Random Forest, Naive Bayes, Stacking Classifier, and Voting Classifier to analyze the dataset, and we achieved 99.96% accuracy for Stacking Classifier and 99.92% accuracy for Voting Classifier. The author of the base paper mentioned using Logistic regression, Linear SVM, and SVM with RBF kernel for analysis and prediction.

Precision, Recall and F1-score :

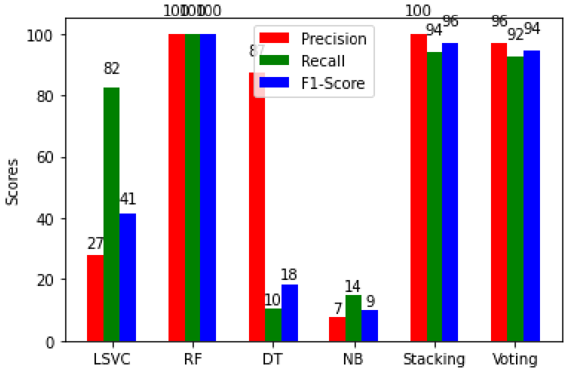


Fig.11: Comparison of all algorithms' metrics

VI. CONCLUSION

In fraud detection, imbalanced datasets are common, and finding a balance between accurately detecting fraudulent transactions and minimizing false positives can be a difficult decision for digital payment companies. We propose a class weight based approach that has shown high accuracy and low false positives on the Paysim dataset. To improve our techniques, we suggest using ensemble techniques to incorporate categorical features and treating the dataset as a time series using algorithms such as CNN. Additionally, creating user-specific models based on past transactional behavior could further enhance our classification quality. These methods have potential to significantly improve fraud detection on the Paysim dataset.

VII. REFERENCES

[1]. A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective - Samaneh Sorournejad, Zojah, Atani et.al - November 2016

[2]. Support Vector machines and malware detection - T.Singh,F.Di Troia, C.Vissagio , Mark Stamp - San Jose State University - October 2015

[3]. Solving the False positives problem in fraud prediction using auto- mated feature engineering - Wedge, Canter, Rubio et.al - October 2017

[4]. Paysim - Synthetic Financial Datasets For Fraud Detection https://www.kaggle.com/ntnu-testimon/paysim1

[5]. A Model for Rule Based Fraud Detection in Telecommunications -Rajani, Padmavathamma - IJERT - 2012

[6]. HTTP Attack detection using n–gram analysis - A. Oza, R.Low, M.Stamp - Computers and Security Journal - September 2014

[7]. V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas, “Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines,” Ore Geol. Rev., vol. 71, pp. 804–818, 2015.

[8]. Y.-J. Lee, Y.-R. Yeh, and H.-K. Pao, “An Introduction to Support Vector Machines,” Psu.edu

Cite this article as :

B. Ravinder Reddy, N. Rajesh, K. V. Anand, G. Srikanth, "Fraud Transaction Detection Approach Using Machine Learning Hybrid Techniques", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 10 Issue 2, pp. 90-96, March-April 2023. Available at doi : https://doi.org/10.32628/IJSRST52310213
Journal URL : https://ijsrst.com/IJSRST52310213