iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71710

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Credit Card Fraud Detection Using Ensemble (Stacking and Voting Classifiers) with Hybrid Techniques

P. Shyam[1], Dr. K. Santhi shree[2]
[1]*Post Graduate Student, M.Tech(CNIS), Department of IT, Jawaharlal Nehru Technological University, Hyderabad, India*
[2]*Professor, Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad, India*

Abstract: ***Credit card fraud remains a critical challenge in the financial industry due to the highly imbalanced nature of fraud detection datasets and the evolving tactics of fraudsters. This study proposes a robust framework for Credit Card Fraud Detection Using Ensemble (Stacking and Voting Classifiers) with Hybrid Techniques, integrating advanced resampling strategies with ensemble learning to enhance the detection of minority fraud cases.We evaluated various machine learning models combined with hybrid oversampling and undersampling methods, including Simple Minority Oversampling Technique(SMOTE)-Tomek, SMOTE Edited Nearest Neighbour(ENN), and Borderline-SMOTE (BSMOTE) with Tomek. Traditional classifiers such as Random Forest (RF), Extreme Gradient Boosting (XGB), and Light Gradient Boosting Machine (LGBM) were benchmarked against ensemble approaches employing stacking and voting classifiers.***

Experimental results demonstrate that Voting Classifier consistently outperforms individual models, achieving the highest F1-score of 0.8634 and AUC of 0.9763 on the CreditCard dataset, and an F1-score of 0.8808 with AUC 0.9961 on the PaySim dataset. The Stacking Classifier also exhibits strong performance, particularly in reducing false positives, evidenced by its superior precision. These findings confirm that integrating hybrid sampling with ensemble models significantly enhances fraud detection capabilities, making the proposed approach effective for real-world financial fraud prevention systems. These results confirm that ensemble classifiers, when combined with appropriate hybrid resampling techniques, can significantly boost fraud detection performance by effectively balancing sensitivity and specificity. The proposed framework showcases the effectiveness of stacking and voting classifiers as part of a hybrid ensemble strategy, providing a reliable, scalable, and adaptable solution for real-world fraud detection systems where early and accurate identification of fraudulent transactions is paramount.

Keywords : Credit Card Fraud Detection, Machine Learning, Hybrid models, Ensemble Methods, Simple Minority Oversampling Technique -Tomek, SMOTE Edited Nearest Neighbour, Borderline-SMOTE, Random Forest, Extreme Gradient Boosting, Light Gradient Boosting Machine, Stacking Classifier, Voting Classifier, Precision, Recall, F1-score, Imbalanced Data, Paysim, Oversampling, UnderSampling

I. INTRODUCTION

The rapid proliferation of digital transactions has amplified the risk of credit card fraud, posing significant threats to financial institutions, merchants, and customers. As fraudsters continue to evolve their tactics, conventional fraud detection methods often struggle to cope with the dynamic, imbalanced, and complex nature of transaction data. Typically, fraudulent transactions represent a minute fraction of the total data, rendering most standard classifiers biased toward the majority (legitimate) class. This imbalance creates a pressing need for sophisticated and intelligent fraud detection systems that can accurately identify rare fraud instances while minimizing false alarms.Recent advancements in machine learning have introduced ensemble methods and hybrid sampling techniques as promising solutions to address the challenges inherent in fraud detection. Ensemble methods, such as stacking and voting classifiers, leverage the strengths of multiple base learners to improve generalization and predictive performance. Meanwhile, hybrid resampling techniques, which combine both oversampling of the minority class and undersampling of the majority class, have shown to be effective in mitigating the skewed class distribution issue.

This study proposes an integrated framework for credit card fraud detection using ensemble classifiers—stacking and voting—enhanced with hybrid resampling techniques, including SMOTE-Tomek, SMOTEENN, and Borderline-SMOTE. The objective is to examine the synergy of these ensemble methods with hybrid data balancing techniques to improve the detection rate of fraudulent transactions while maintaining high overall classification accuracy.

©IJRASET: All Rights are Reserved

Experiments were conducted on two benchmark datasets—CreditCard and PaySim—where various models, including Random Forest (RF), Extreme Gradient Boosting (XGB), and LightGBM (LGBM), were assessed individually and within ensemble frameworks. The results clearly demonstrate that stacking and voting classifiers, when integrated with hybrid sampling strategies, significantly enhance fraud detection metrics such as precision, recall, F1-score, and ROC AUC, particularly for the minority fraud class. This research underscores the potential of combining ensemble learning with hybrid resampling as a robust approach to tackling the complexities of real-world credit card fraud detection, offering scalable and high-performance solutions for financial institutions.

II. LITERATURE SURVEY

The problem of class imbalance in credit card fraud detection has garnered significant attention in recent years. Various sampling techniques have been proposed to address the challenges posed by skewed datasets.

Alamri and Ykhlef proposed a hybrid sampling method combining Tomek links, BIRCH clustering, and Borderline-SMOTE to handle imbalanced credit card data. Their method initially applies Tomek links to remove majority class instances that are borderline or noisy. This is followed by BIRCH clustering to group similar instances and finally, Borderline-SMOTE is used to oversample the minority class within these clusters. The approach showed superior performance compared to baseline methods, achieving an F1-score of 85.20% using a Random Forest classifier [1].

ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection: Enhanced F1-scores and reduced false positives by using adaptive oversampling strategies (Tomek Links, ADASYN) combined with Random Forest and Balanced Random Forest classifiers.[2]

A Behavior-cluster Based Imbalanced Classification Method for Credit Card Fraud Detection: Achieved high precision and recall by applying hybrid resampling methods for imbalanced fraud detection using SMOTE, Tomek Links, and XGBoost.[3]

SMOTE-NCL: A re-sampling method with filter for network intrusion detection: Effectively handled imbalanced credit card fraud datasets, achieving higher accuracy and AUC-ROC curves using SMOTE, Tomek Links, and Random Forest.[4]

NUS: Noisy-Sample-Removed Undersampling Scheme for Imbalanced Classification and Application to Credit Card Fraud Detection: Improved classifier performance using hybrid SMOTE-ENN and Tomek Links for handling imbalanced data with Balanced Random Forest.[5]

Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset: Demonstrated significant improvement in precision and recall for credit card fraud detection using hybrid approaches (Tomek Links, SMOTE, Random Forest, SMOTEENN, XGBoost).[6]

Detection of fraudulent credit card transactions: A comparative analysis of data sampling and classification techniques: Enhanced fraud detection by combining SMOTE with classifiers like Random Forest and XGBoost, and using Tomek Links and Borderline-SMOTE.[7]

Kaur and Gosain explored the effectiveness of handling class imbalance in fraud detection using various sampling methods like SMOTE, random oversampling, and undersampling. They applied these techniques with classifiers including Naive Bayes and Decision Tree. The study found that SMOTE significantly improves the model's ability to detect minority class instances. It highlighted SMOTE's superior performance over traditional undersampling in imbalanced scenarios. [8].

This study compared SMOTE and ADASYN oversampling methods with classifiers like Random Forest and Decision Tree for credit card fraud detection. The results indicated that oversampling improved the classifiers' sensitivity to fraudulent transactions. Among the methods, SMOTE achieved higher recall, making it more effective for minority class detection. The authors emphasized the importance of balancing techniques for real-world fraud data. [9].

Mahesh et al. proposed a comparative framework for fraud detection using classifiers such as Random Forest, Logistic Regression, and KNN in conjunction with SMOTE and undersampling. Their results showed that the Random Forest model with SMOTE had the highest accuracy. The research underlined the benefit of combining oversampling with ensemble methods to address data imbalance. It reinforced the role of preprocessing in fraud detection pipelines. [10]. Rtayli focused on deep learning-based models for detecting credit card fraud in highly imbalanced datasets. Using a deep neural network (DNN), the model achieved high detection accuracy without extensive sampling or balancing techniques. The study demonstrated that deep learning can inherently handle complex fraud patterns. This work supports the growing shift towards neural models in fraud analytics. [11].

This research applied various machine learning algorithms such as Logistic Regression, Decision Tree, Naive Bayes, SVM, and Random Forest to a credit card fraud dataset.

Random Forest achieved the best overall performance in detecting fraudulent transactions. The study emphasized the importance of model selection and feature engineering. It concluded that ensemble techniques are generally more reliable for fraud detection.[12]. Li and Xie introduced a novel behavior-based clustering method followed by classification using models like SVM and Decision Tree. The approach grouped similar transaction behaviors before classification to improve accuracy. Their method enhanced the detection of fraudulent transactions in imbalanced datasets. The study suggested behavior clustering as a valuable preprocessing step for fraud detection. [13].

Esenogho et al. utilized This work proposed an ensemble of neural networks integrated with feature engineering for fraud detection. A voting mechanism among the models was employed to enhance reliability. The ensemble outperformed single classifiers in precision and recall, particularly for fraud cases. Their approach showcased the effectiveness of combining deep learning with ensemble strategies. [14]. Yi and colleagues developed ASN-SMOTE, a variant of SMOTE designed to generate more diverse and representative synthetic samples. When tested with Random Forest and XGBoost, ASN-SMOTE outperformed traditional oversampling methods. It improved minority class prediction while reducing overfitting. This innovation enhances model generalization on imbalanced datasets. [15].

Ullastres and Latifi evaluated the performance of multiple models including Logistic Regression, Random Forest, XGBoost, and SVM on a credit card fraud dataset. Their analysis highlighted the strength of ensemble models like XGBoost in detecting fraud. The study emphasized the role of proper feature selection and data balancing in improving model outcomes. XGBoost showed superior precision and recall compared to baseline classifiers. [16]. Zhu et al. proposed a Noisy-sample-removed Undersampling (NUS) method to improve fraud detection. By removing noisy samples before undersampling, the model achieved higher detection accuracy. The technique proved particularly effective with classifiers like SVM and neural networks. Their findings suggest that cleaning the data before balancing enhances classifier robustness. [17].

Lopez-Rojas et al. developed PaySim, a financial mobile money simulator for fraud detection, providing a realistic dataset for evaluating different fraud detection techniques [18]. Arfeen and Khan conducted an empirical analysis of machine learning algorithms for detecting fraudulent electronic fund transfers, reinforcing the importance of algorithm selection in handling imbalanced data [19]. Mondal et al. explored handling imbalanced data for credit card fraud detection, emphasizing the significance of integrating sampling techniques with advanced classifiers [20].

III. DATASET

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.We used a synthetic credit card transaction dataset with a significant class imbalance between fraud and non-fraud transactions.The dataset considered having a significant number of records with non fraud and very few number of fraud data compared to non fraud data. It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.[21]

Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	-1.3598	-0.0728	2.53635	1.37816	-0.3383	0.46239	0.2396	0.0987	0.36379	0.09079	-0.5516	-0.6178	-0.9914	-0.3112	1.46818	-0.4704	0.20797	0.02579	0.40399	0.25141	-0.0183	0.27784	-0.1105	0.06693	0.12854	-0.1891	0.13356	-0.0211	149.62	0
0	1.19186	0.26615	0.16648	0.44815	0.06002	-0.0824	-0.0788	0.0851	-0.2554	-0.167	1.61273	1.06524	0.4891	-0.1438	0.63556	0.46392	-0.1148	-0.1834	-0.1458	-0.0691	-0.2258	-0.6387	0.10129	-0.3398	0.16717	0.12589	-0.009	0.01472	2.69	0
1	-1.3584	-1.3402	1.77321	0.37978	-0.5032	1.8005	0.79146	0.24768	-1.5147	0.20764	0.6245	0.06608	0.71729	-0.1659	2.34586	-2.8901	1.10997	-0.1214	-2.2619	0.52498	0.248	0.77168	0.90941	-0.6893	-0.3276	-0.1391	-0.0554	-0.0598	378.66	0
1	-0.9663	-0.1852	1.79299	-0.8633	-0.0103	1.2472	0.23761	0.37744	-1.387	-0.055	-0.2265	0.17823	0.50776	-0.2879	-0.6314	-1.0596	-0.6841	1.96578	-1.2326	-0.208	-0.1083	0.00527	-0.1903	-1.1756	0.64738	-0.2219	0.06272	0.06146	123.5	0
2	-1.1582	0.87774	1.54872	0.40303	-0.4072	0.09592	0.59294	-0.2705	0.81774	0.75307	-0.8228	0.5382	1.34585	-1.1197	0.17512	-0.4514	-0.237	-0.0382	0.80349	0.40854	-0.0094	0.79828	-0.1375	0.14127	-0.206	0.50229	0.21942	0.21515	69.99	0
2	-0.426	0.96052	1.14111	-0.1683	0.42099	-0.0297	0.4762	0.26031	-0.5687	-0.3714	1.34126	0.35989	-0.3581	-0.1371	0.51762	0.40173	-0.0581	0.06865	-0.0332	0.08497	-0.2083	-0.5598	-0.0264	-0.3714	-0.2328	0.10591	0.25384	0.08108	3.67	0
4	1.22966	0.141	0.04537	1.20261	0.19188	0.27271	-0.0052	0.08121	0.46496	-0.0993	-1.4169	-0.1538	-0.7511	0.16737	0.05014	-0.4436	0.00282	-0.612	-0.0456	-0.2196	-0.1677	-0.2707	-0.1541	-0.7801	0.75014	-0.2572	0.03451	0.00517	4.99	0
7	-0.6443	1.41796	1.07438	-0.4922	0.94893	0.42812	1.12063	-3.8079	0.61537	1.24938	-0.6195	0.29147	1.75796	-1.3239	0.68613	-0.0761	-1.2221	-0.3582	0.3245	-0.1567	1.94347	-1.0155	0.0575	-0.6497	-0.4153	-0.0516	-1.2069	-1.0853	40.8	0
7	-0.8943	0.28616	-0.1132	-0.2715	2.6696	3.72182	0.37015	0.85108	-0.392	-0.4104	-0.7051	-0.1105	-0.2863	0.07436	-0.3288	-0.2101	-0.4998	0.11876	0.57033	0.05274	-0.0734	-0.2681	-0.2042	1.01159	0.3732	-0.3842	0.01175	0.1424	93.2	0

Table 1 : Credit card dataset

PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.This synthetic dataset is scaled down 1/4 of the original dataset and it is created just for Kaggle.We start with loading the dataset and explore the data with considering which type of data is available and how many types of transacttions are done and also considering by which methods.Then preprocess the data and select the training and testing sets at 0.8 and 0.2 of the total data.Below are the features of paysim dataset,[22]

- Step - maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).
- Type - CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.
- Amount -amount of the transaction in local currency.
- nameOrig - customer who started the transaction
- oldbalanceOrg - initial balance before the transaction newbalanceOrig - new balance after the transaction.
- nameDest - customer who is the recipient of the transaction
- oldbalanceDest - initial balance recipient before the transaction. Note that there is not information for customers that start with M (Merchants).
- newbalanceDest - new balance recipient after the transaction. Note that there is not information for customers that start with M (Merchants).
- isFraud - This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers accounts and try to empty the funds by transferring to another account and then cashing out of the system.
- isFlaggedFraud - The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.

step	type	amount	nameOrig	oldbalanc	newbalan	nameDest	oldbalanc	newbalan	isFraud	isFlaggedFraud
1	PAYMENT	9839.64	C1231006	170136	160296.4	M1979787	0	0	0	0
1	PAYMENT	1864.28	C1666544	21249	19384.72	M2044282	0	0	0	0
1	TRANSFER	181	C1305486	181	0	C5532640	0	0	1	0
1	CASH_OU	181	C8400836	181	0	C3899701	21182	0	1	0
1	PAYMENT	11668.14	C2048537	41554	29885.86	M1230701	0	0	0	0
1	PAYMENT	7817.71	C9004563	53860	46042.29	M5734872	0	0	0	0

Table 2 : Paysim Dataset

IV. PROPOSED METHODOLOGY

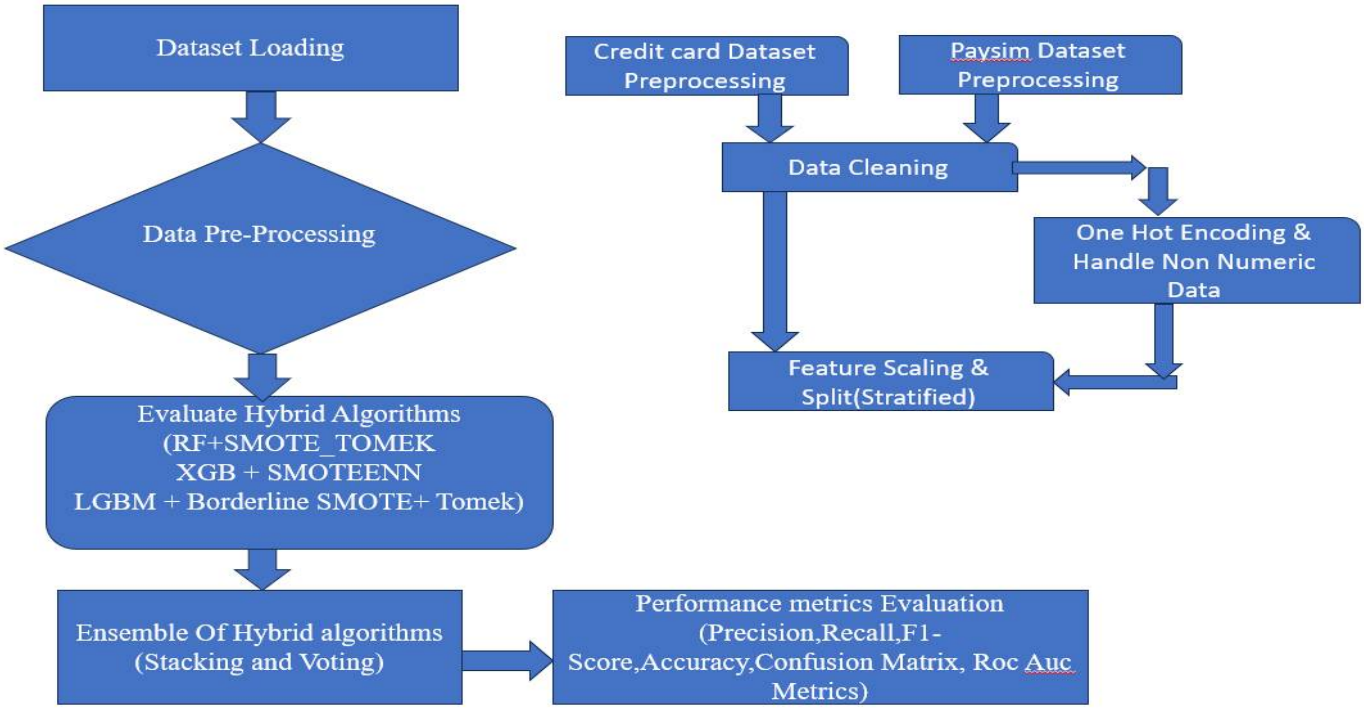


Figure 1 : Proposed Architecture

A. Dataset Loading

The initial step consists of loading two datasets one after the other and generate dataset insights that includes shape of the dataset, memory usage of the dataset, class distribution, etc.

B. Data Preprocessing

Perform preprocessing separately for each dataset ("CreditCard" and "PaySim"). Ensure that *all* preprocessing steps maintain the original stratified class distribution.The 1st dataset(European) with records of 284,807 is loaded and undergoes Perform standard data cleaning operations (handling missing values, outliers, etc.). For 2nd dataset(Paysim) is loaded and it handles missing values and the dataset is then pre-processed to prepare it for machine learning algorithms. Irrelevant features such as origin and destination identifiers are dropped. Categorical features, specifically the transaction type, are converted to numerical values using one hot encoding.

C. .Train -test data

The pre-processed data is split into training and testing sets using an 80-20 split ratio. Stratifaction is applied during the split to maintain the class distribution in both sets, which is crucial for ensuring that the models trained on this data will generalize well to unseen data and remove underfitting and overfitting issues usually occurs when balancing datasets.

D. Hybrid and Ensemble Methods

It is a three stage pipeline architecture consisting of Under-sampling techniques , Over-sampling Techniques and a Classifier which the acts as base learners for ensemble techniques like Stacking and Voting classifiers.

1) RF + Smote Tomek : SMOTE (Synthetic Minority Over-sampling Technique) Generates synthetic examples of the minority class (fraudulent transactions) by interpolating between existing ones. Tomek Links Cleans the overlapping data points between classes by removing Tomek link pairs (samples that are very close but from different classes), reducing noise and class ambiguity. Random Forest (RF) is an ensemble method that builds multiple decision trees and merges their outputs (via majority voting) to improve classification performance and reduce overfitting.

2) XGBoost + SMOTEENN : SMOTEENN (SMOTE + Edited Nearest Neighbors) SMOTE generates synthetic minority class samples. ENN removes ambiguous or misclassified examples using a k-nearest neighbor approach. XGBoost (Extreme Gradient Boosting): A powerful gradient boosting framework that builds sequential trees to reduce bias and improve performance. It's regularized, scalable, and handles missing values well.

3) LightGBM + Borderline-SMOTE+Tomek : Borderline-SMOTE Unlike regular SMOTE, this method focuses on generating synthetic samples only near the decision boundary—where misclassification is most likely. It avoids generating data in safe zones (which are already well-classified). Tomek Links Cleans the overlapping data points between classes by removing Tomek link pairs (samples that are very close but from different classes), reducing noise and class ambiguity. LightGBM (Light Gradient Boosting Machine) is a gradient boosting algorithm optimized for speed and efficiency. It uses histogram-based decision trees and is suitable for large datasets.

4) STACKING : We employed Stacking to combine the strengths of multiple classifiers

Base learners: Random Forest, XGBoost, LightGBM

Meta-learner : Logistic Regression

In stacking, the base models predict outcomes independently, and their outputs are used as input features to train the meta-learner, which gives the final prediction. Stacking helps leverage the diversity of different models to improve generalization and fraud detection.

5) VOTING : We also implemented a soft voting ensemble Combines RF, XGB, and LGBM predictions by averaging their predicted probabilities Final class is chosen based on the highest average probability Voting provides a simple yet effective ensemble strategy, especially when base models perform comparably.

E. Evaluate Model Performance :

Each trained model is evaluated based on several performance metrics:

- Accuracy : The ratio of correctly predicted instances to the total instances.

$$Accuracy=(TP+TN)/(TP+TN+FP+FN)$$

Where:

- TP: True Positives (correctly predicted positive instances)
- TN: True Negatives (correctly predicted negative instances)
- FP: False Positives (incorrectly predicted positive instances)
- FN: False Negatives (incorrectly predicted negative instances)
-

- Precision (Fraud Class): Positive predicted value measures the proportion of correctly predicted positive instances out of all instances predicted as positive.

$$Precision=TP/(FP+TP)$$

- Recall (Fraud Class): Sensitivity or True Positive Rate measures the proportion of correctly predicted positive instances out of all actual positive instances.

$$Recall=TP/(FN+TP)$$

- F1-Scare (Fraud Class): harmonic mean of precision and recall, providing a single metric that balances both the precision and recall of the model.

$$F1\text{-}Score=2\times((Precision\times Recall)/(Precision+Recall))$$

- ROC CURVE: The Receiver Operating Characteristic (ROC) Curve is a graphical plot that illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied. It ranges from 0.0 to 1.0
  - AUC = 1.0: Perfect classifier
  - AUC = 0.5: No better than random guessing
  - AUC > 0.8: Good performance
- Confusion Matrix: A confusion matrix is a performance measurement tool for classification problems. It allows visualization of the model's performance by showing true vs. predicted classifications.

V. EXPERIMENTAL RESULTS

We evaluated several hybrid and ensemble machine learning techniques for fraud detection using two highly imbalanced datasets: the CreditCard dataset and the PaySim dataset. Our experimentation included combinations such as RF + SMOTE-Tomek, XGB + SMOTEENN, XGB + BSMOTE + Tomek, and advanced ensemble methods like Stacking and Voting Classifiers.

Performance metrics for RF + SMOTE – Tomek(Hybrid 1) **CreditCard :** Achieved strong fraud detection with F1-score of 0.8482 and high ROC AUC of 0.9782, showing reliable balance between precision and recall. **PaySim:** Delivered better recall (0.8421) and an excellent AUC of 0.9875, proving effective at identifying rare fraud cases.

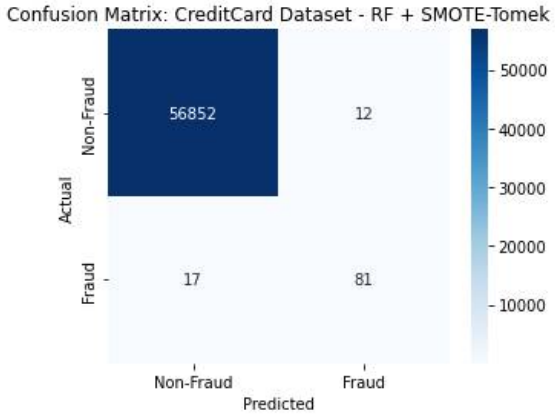


Figure 2 : confusion matix of Credit card dataset(Hybrid 1)

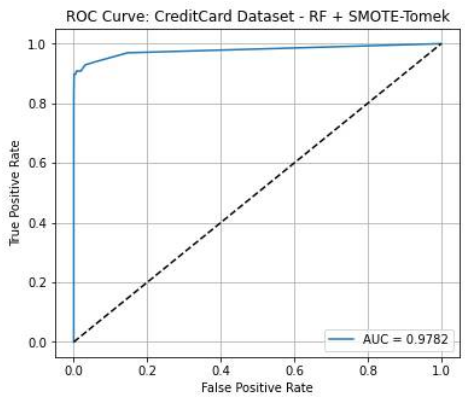


Figure 3 : ROC Curve of Credit card dataset(Hybrid 1)

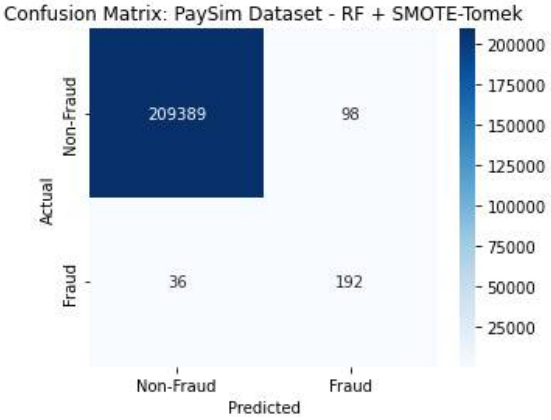


Figure 4 : Confusion matrix of Paysim dataset(Hybrid 1)

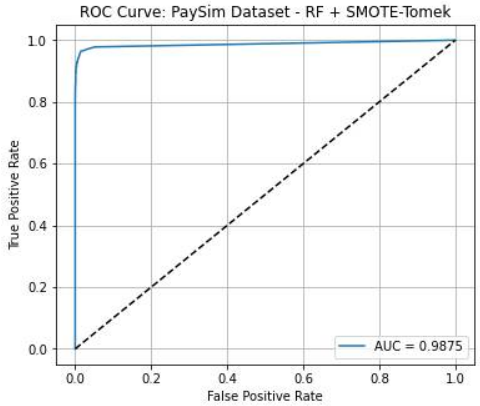


Figure 5 : ROC Curve of Paysim dataset(Hybrid 1)

Performance Metrics for - XGB + SMOTEENN(Hybrid 2) CreditCard: High recall (0.8571) but low precision (0.6131) indicates more false positives, despite good AUC (0.9804). PaySim: Extremely high recall (0.9035) but poor precision (0.2068), leading to many false alarms; best suited when catching all frauds is critical.

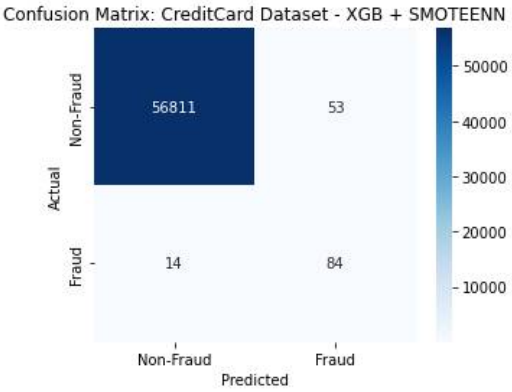


Figure 6 : Confusion matrix for Credit card dataset(Hybrid 2)

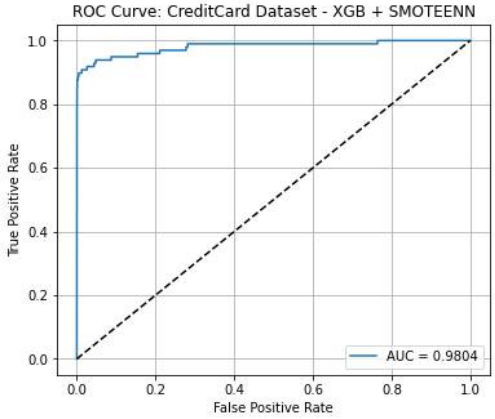


Figure 7 : ROC Curve for Credit card dataset(Hybrid 2)

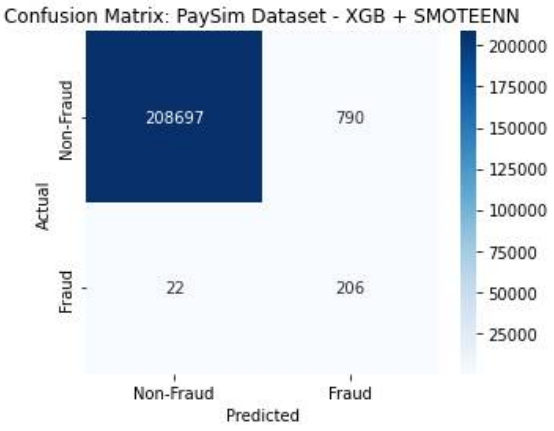


Figure 8 : Confusion matrix for Paysim dataset(Hybrid 2)

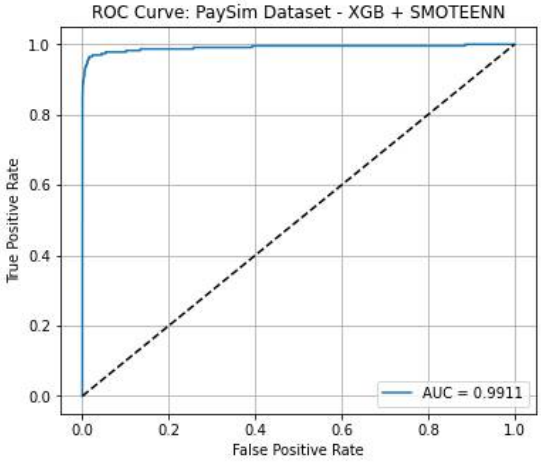


Figure 9 : ROC Curve for Paysim dataset(Hybrid 2)

Performance Metrics for XGB + BSMOTE + Tomek(Hybrid 3): **CreditCard:** Maintained solid performance with F1-score of 0.8316 and balanced metrics, indicating stable fraud detection. **PaySim:** Delivered a balanced fraud detection rate with precision of 0.8873 and F1-score of 0.8571, suggesting high reliability.

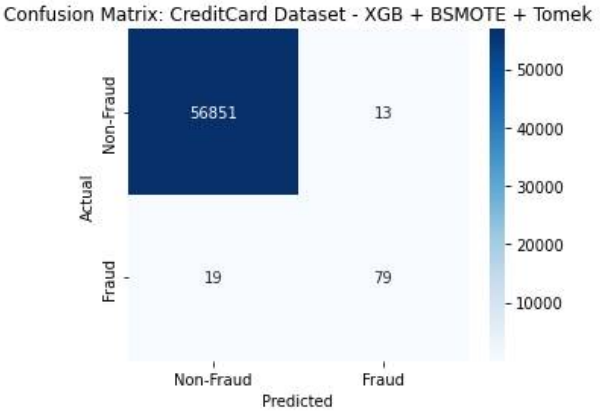


Figure 10 : Confusion matrix for Credit card dataset(Hybrid 3)

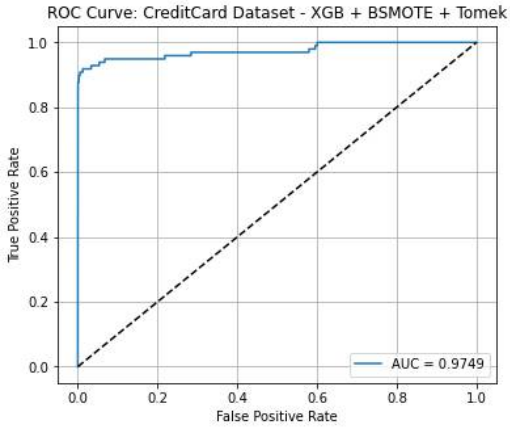


Figure 11 : ROC Curve for Credit card dataset(Hybrid 3)

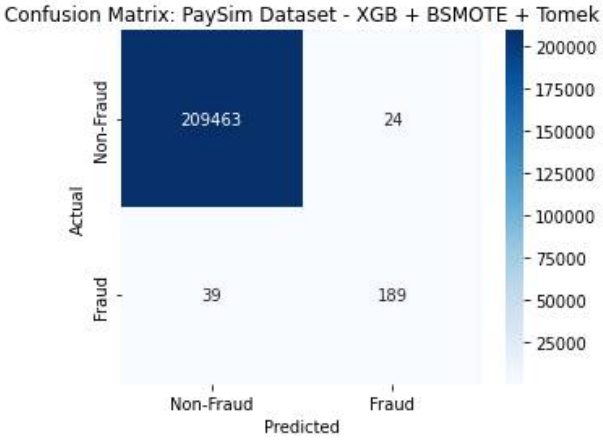


Figure 12 : Confusion matrix for Paysim dataset(Hybrid 3)

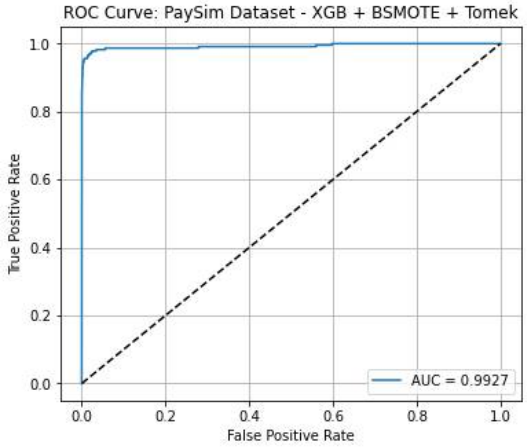


Figure 13 : ROC Curve for Paysim dataset(Hybrid 3)

Performance Metrics for STACKING : **CreditCard:** Showed robust performance with highest fraud precision (0.9268) and strong F1-score (0.8444), ideal for reducing false positives. **PaySim:** Achieved exceptional fraud precision (0.9941) and AUC (0.9972), demonstrating its strength in both detection and reliability.

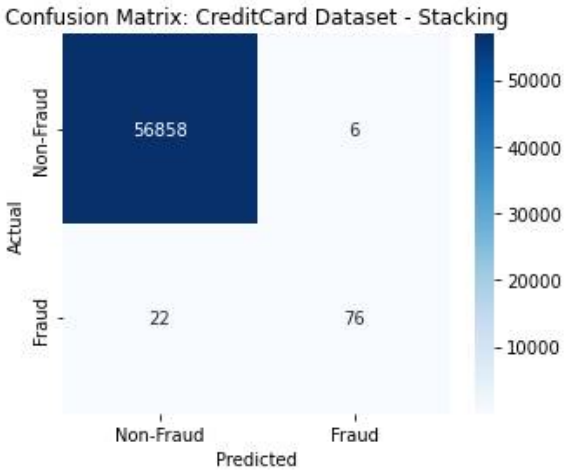


Figure 14 : Confusion matrix for Credit card dataset(stacking)

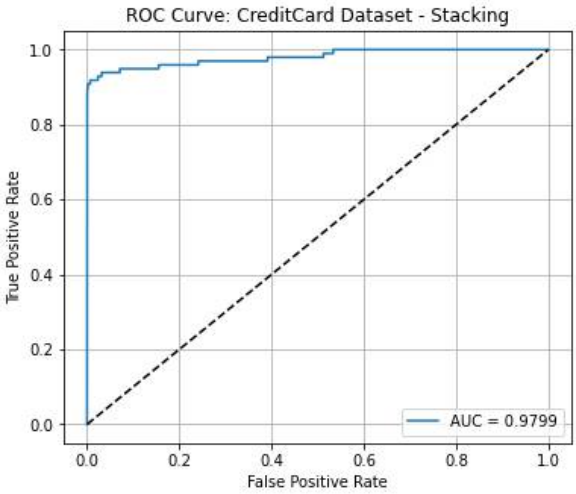


Figure 15 : ROC Curve for Credit card dataset(Stacking)

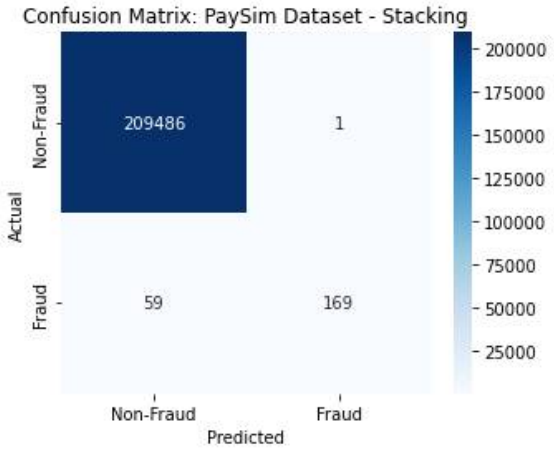


Figure 16 : Confusion matrix for Paysim card dataset(stacking)

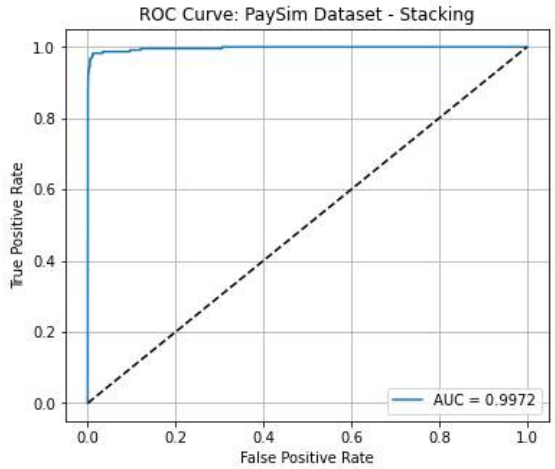


Figure 17 : ROC Curve for Paysim dataset(Stacking)

Performance Metrics for VOTING : **CreditCard:** Balanced fraud recall (0.8061) and highest F1-score (0.8634), proving to be the best overall performer. **PaySim:** Offered top-tier results with precision of 0.9891 and F1-score of 0.8808, making it highly suitable for real-world deployment.

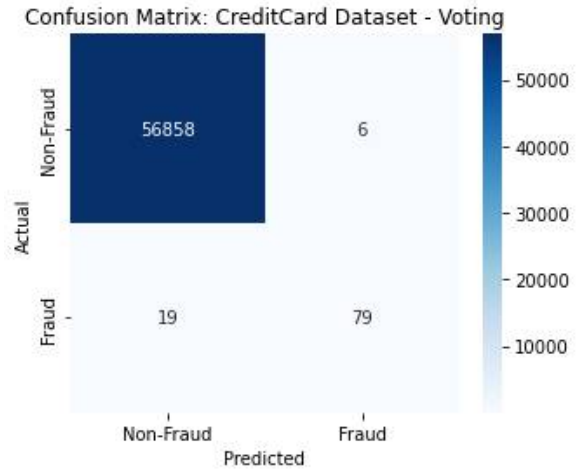


Figure 18 : Confusion matrix for Credit card dataset(Voting)

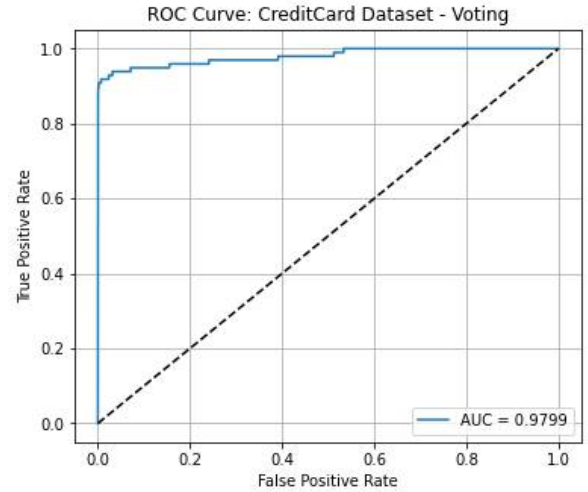


Figure 19 : ROC Curve for Credit card dataset(Voting)

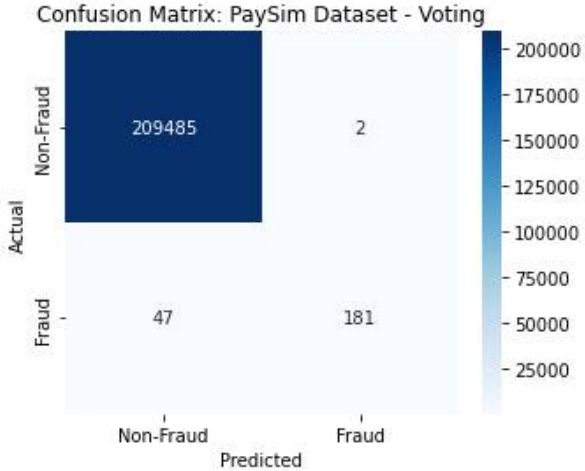


Figure 20 : Confusion matrix for Credit card dataset(Voting)

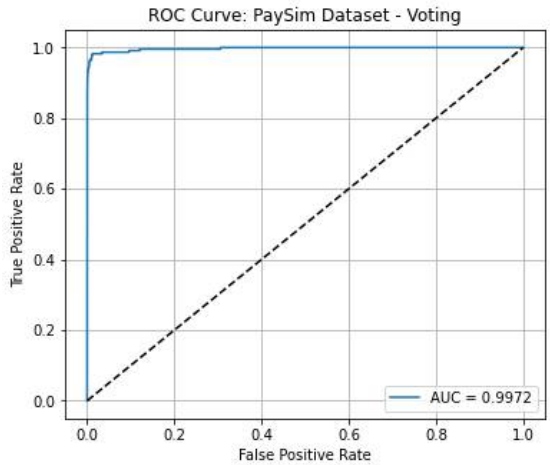


Figure 21 : ROC Curve for Credit card dataset(Voting)

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

For the CreditCard dataset, the best performance was achieved by the Voting Classifier, which yielded a fraud precision of 0.9294, recall of 0.8061, and an F1-score of 0.8634, with an accuracy of 99.96% and ROC AUC of 0.9799, indicating robust fraud detection capability while maintaining high overall performance. Similarly, the Stacking model delivered competitive results with a fraud precision of 0.9268 and an F1-score of 0.8444. For the PaySim dataset, the Voting Classifier again outperformed other models, achieving fraud precision of 0.9891, recall of 0.7939, and F1-score of 0.8808, with a ROC AUC of 0.9972. Notably, although XGB + SMOTEENN showed high recall (0.9035) on the PaySim dataset, it suffered from poor precision (0.2068), leading to lower F1-scores. Overall, ensemble approaches, particularly Voting and Stacking, demonstrated superior effectiveness in balancing high precision and recall, making them well-suited for fraud detection tasks on imbalanced datasets.

VI. CONCLUSIONS

The proposed methedology demonstrates the effectiveness of combining ensemble learning with advanced resampling strategies to address the challenges posed by highly imbalanced fraud detection datasets. Two benchmark datasets—CreditCard and PaySim—were utilized to evaluate the performance of various hybrid models.Across both datasets, ensemble methods such as Stacking and Voting classifiers consistently outperformed individual hybrid approaches (e.g., Random Forest + SMOTE-Tomek, XGBoost + SMOTEENN) in terms of precision, recall, F1-score, and ROC AUC, particularly for the minority fraud class. Notably, on the CreditCard dataset, the Voting classifier achieved the highest fraud F1-score of 0.8634, with a strong balance between precision (0.9294) and recall (0.8061).

Similarly, the PaySim dataset results revealed the Voting classifier as the top performer with an exceptional fraud F1-score of 0.8808, precision of 0.9891, and recall of 0.7939, indicating a robust ability to correctly identify fraudulent transactions while minimizing false positives.The use of hybrid resampling techniques such as SMOTE-Tomek, SMOTEENN, and BSMOTE + Tomek significantly contributed to improving the detection rates of fraud cases by generating synthetic examples and cleaning noisy data, thus aiding classifiers in learning more discriminative patterns. Furthermore, the ensemble frameworks effectively leveraged the strengths of base learners to build more generalized and accurate models.

Overall, the results validate that ensemble methods combined with hybrid sampling techniques provide a powerful and reliable solution for credit card fraud detection, offering high predictive performance and addressing the critical issue of class imbalance. This approach not only enhances fraud detection capabilities but also reduces operational risk for financial institutions by enabling faster and more accurate identification of fraudulent activities.

REFERENCES

[1] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. A. Khalid, ‘‘Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset,’’ in Proc. IEEE 16th Int. Conf. Control Autom. (ICCA), Oct. 2020, pp. 803–808, doi: 10.1109/ICCA51439.2020.9264517.

[2] Y. Zhang, J. Wang, and J. Ma, “ASN-SMOTE: A synthetic minority oversampling method with adaptive qualified synthesizer selection,” Neural Comput. Appl., vol. 34, no. 12, pp. 9939–9952, Jun. 2022, doi: 10.1007/s40747-021-00638-w.

[3] Y. Wang, H. Wang, and Y. Chen, “A behavior-cluster based imbalanced classification method for credit card fraud detection,” in Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), Nov. 2019, pp. 2397–2400, doi: 10.1145/3352411.3352433.

[4] L. Douzas and F. Bacao, “SMOTE-NCL: A re-sampling method with filter for network intrusion detection,” in Proc. IEEE Int. Conf. Comput. Commun. (COMP COMM), Dec. 2016, pp. 1–6, doi: 10.1109/COMPCOMM.2016.7924886.

[5] Q. Wang, Y. Zhang, and X. Liu, “NUS: Noisy-sample-removed undersampling scheme for imbalanced classification and application to credit card fraud detection,” IEEE Trans. Comput. Soc. Syst., vol. 11, no. 1, pp. 123–134, Mar. 2024, doi: 10.1109/TCSS.2023.3243925.

[6] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. A. Khalid, “Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset,” in Proc. IEEE 16th Int. Conf. Control Autom. (ICCA), Oct. 2020, pp. 803–808, doi: 10.1109/ICCA51439.2020.9264517.

[7] S. A. S. Sadiq, M. A. Hussain, and S. A. Khan, “Detection of fraudulent credit card transactions: A comparative analysis of data sampling and classification techniques,” J. Phys. Conf. Ser., vol. 1742, no. 1, Art. no. 012072, 2021, doi: 10.1088/1742-6596/2161/1/012072.

[8] P. Kaur and A. Gosain, ‘‘Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise,’’ in Advances in Intelligent Systems and Computing. Singapore: Springer, 2017, pp. 23–30, doi: 10.1007/978-981-10-6602-3_3.

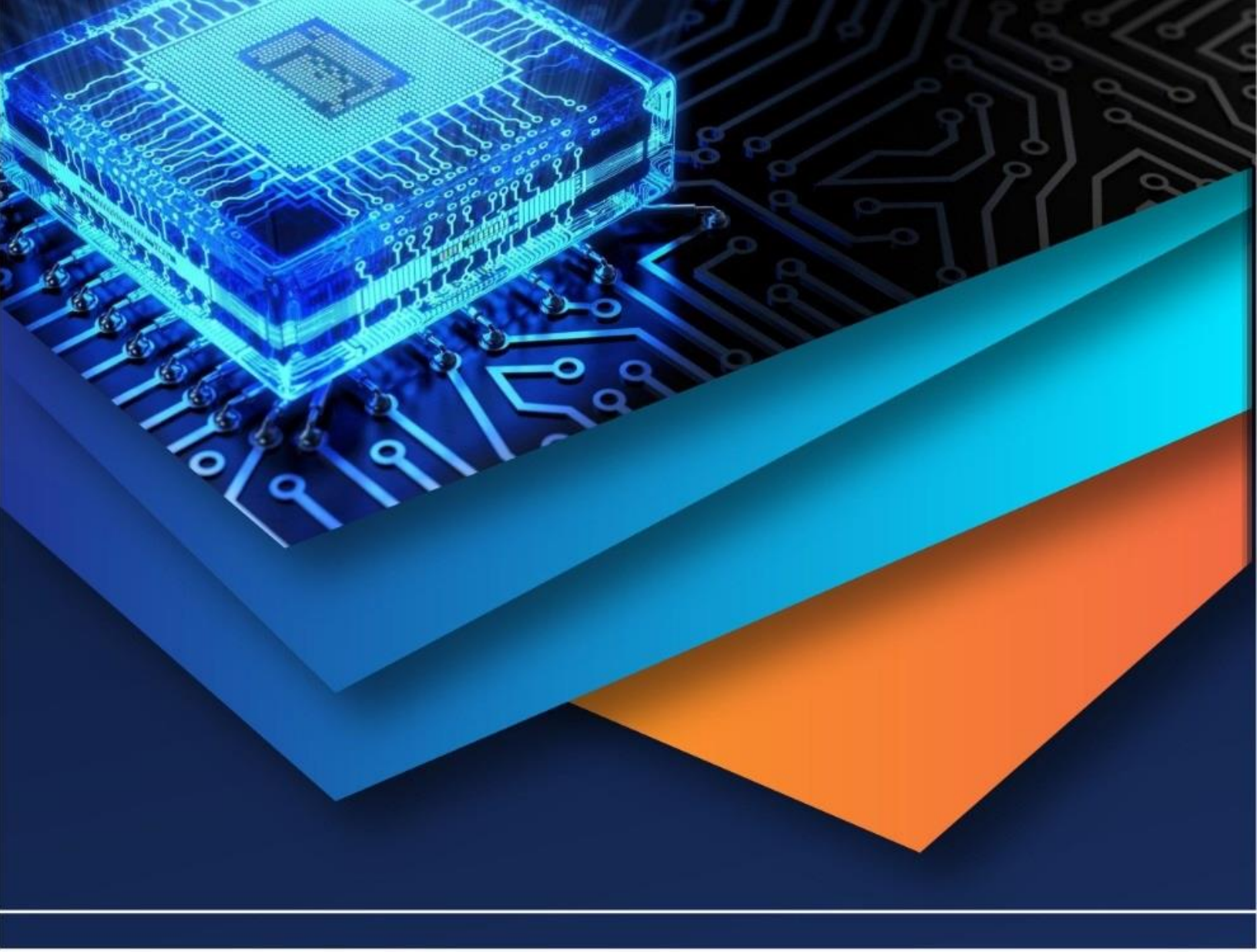
[9] R. Qaddoura and M. M. Biltawi, ‘‘Improving fraud detection in an imbalanced class distribution using different oversampling techniques,’’ in Proc. Int. Eng. Conf. Electr., Energy, Artif. Intell. (EICEEAI), Nov. 2022, pp. 1–5, doi: 10.1109/EICEEAI56378.2022.10050500.

[10] K. Praveen Mahesh, S. Ashar Afrouz, and A. Shaju Areeckal, ‘‘Detection of fraudulent credit card transactions: A comparative analysis of data sampling and classification techniques,’’ in Proc. J. Phys., Conf., Jan. 2022, vol. 2161, no. 1, Art. no. 012072, doi: 10.1088/1742-6596/2161/1/012072.

[11] N. Rtayli, ‘‘An efficient deep learning classification model for predicting credit card fraud on skewed data,’’ J. Inf. Secur. Cybercrimes Res., vol. 5, no. 1, pp. 57–71, Jun. 2022, doi: 10.26735/tlyg7256.

[12] S. O. Akinwamide, ‘‘Prediction of fraudulent or genuine transactions on credit card fraud detection dataset using machine learning techniques,’’ Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 6, pp. 5061–5071, Jun. 2022, doi: 10.22214/ijraset.2022.44962.

[13] Q. Li and Y. Xie, ‘‘A behavior-cluster based imbalanced classification method for credit card fraud detection,’’ in Proc. 2nd Int. Conf. Data Sci. Inf. Technol. New York, NY, USA: ACM, Jul. 2019, pp. 134–139, doi: 10.1145/3352411.3352433.
[14] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, ‘‘A neural network ensemble with feature engineering for improved credit card fraud detection,’’ IEEE Access, vol. 10, pp. 16400–16407, 2022, doi: 10.1109/ACCESS.2022.3148298.
[15] X. Yi, Y. Xu, Q. Hu, S. Krishnamoorthy, W. Li, and Z. Tang, ‘‘ASN-SMOTE: A synthetic minority oversampling method with adaptive qualified synthesizer selection,’’ Complex Intell. Syst., vol. 8, no. 3, pp. 2247–2272, Jun. 2022, doi: 10.1007/s40747-021-00638-w.
[16] E. F. Ullastres and M. Latifi, ‘‘Credit card fraud detection using ensemble learning algorithms MSc research project MSc data analytics,’’ M.S. thesis, Nat. College Ireland, Dublin, Ireland, May 2022.
[17] H. Zhu, M. Zhou, G. Liu, Y. Xie, S. Liu, and C. Guo, ‘‘NUS: Noisy-sample-removed undersampling scheme for imbalanced classification and application to credit card fraud detection,’’ IEEE Trans. Intell. Transp. Syst., vol. 23, no. 9, pp. 17601–17611, Sep. 2022, doi: 10.1109/TITS.2022.3165638.
[18] E. G. Lopez-Rojas, A. Elmir, and S. Axelsson, ‘‘PaySim: A financial mobile money simulator for fraud detection,’’ in Proc. 28th Eur. Modeling Symp. (EMS), Oct. 2014, pp. 249–255, doi: 10.1109/EMS.2014.50.
[19] A. Arfeen and F. H. Khan, ‘‘Empirical analysis of machine learning algorithms for detecting fraudulent electronic fund transfers,’’ J. Artif. Intell. Data Sci., vol. 1, no. 2, pp. 71–80, Dec. 2021, doi: 10.47693/jaids.v1i2.50.
[20] H. Mondal, ‘‘Handling imbalanced data for credit card fraud detection using various algorithms: An empirical study,’’ in Proc. 2nd Int. Conf. Smart Technol. Intell. Syst. (STIS), Nov. 2022, pp. 1–8, doi: 10.1109/STIS57120.2022.10000935.
[21] Dataset “Credit Card Fraud Detection Anonymized European Card Holders transactions labeled as fraudulent or genuine” https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
22. Dataset 2 : “Synthetic Financial Datasets For Fraud Detection Synthetic datasets generated by the PaySim mobile money simulator”https://www.kaggle.com/datasets/ealaxi/paysim1/data



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)