ISSN: 2366-1313

Identity Theft Detection Using SMOTE Technique for Credit-Card Fraudulent Transaction System

Mrs. Srilatha Komakula [1] , **Dr. M. Jagadeeshwar** [2]

[1] Research Scholar, Department of Computer Science, Chaitanya (Deemed to be University), Warangal, Telangana State, India

[2] Professor, Department of Computer Science, Chaitanya (Deemed to be University), Warangal, Telangana State, India

Author Email: srilatha.kom@gmail.com

Abstract: The usage of credit cards has increased significantly as the world becomes increasingly digital and financial transactions happen online. The increasing amount of fraud associated with it causes financial institutions to endure huge losses. We must, therefore, investigate and distinguish between fraudulent and non-fraudulent transactions. We planned to apply the full model training process from start to finish for this investigation. The outcome will be the acquisition of the most effective model capable of differentiating regular transactions from abnormal ones. Credit card fraud is detected using machine learning algorithms, but no systems that are particularly successful at detecting it have been produced so far. The relatively new field of deep learning has been used to solve difficult problems across numerous domains. The purpose of this article is to examine various machine learning models for detecting credit card fraud. We compare each model's performance and output. The best possible performance is possible when the SMOTE technique is used. Undersampling the majority (normal) class is a useful strategy for increasing a classifier's sensitivity to the minority class. This work illustrates that our method of oversampling the minority (abnormal) class and undersampling the majority (normal) class together can increase the classifier's performance more than just undersampling the majority class.

Keywords: Classification algorithms; Machine Learning; Deep learning; Fraudulent Credit Card; Identity theft

I. INTRODUCTION

Internet users now buy items online for various purposes, including information exchange, social interaction, entertainment, and financial transactions. However, if third parties gain private information, they can use it to open new accounts, control credit accounts, or apply for government benefits. Machine learning (ML) algorithms can help detect identity theft by acquiring knowledge from past fraudulent activities and identifying them in subsequent transactions. This technology allows for a safer online shopping experience for consumers [1, 2]. In the absence of significant human intervention, analytical algorithms and systems that "learn" patterns from specimens and data points are categorized under the umbrella term "machine learning." Machine learning is a generic phrase. Since machine learning can process information much faster than manual investigation and is much more beneficial to scale when dealing with larger associations and large amounts of data, it is essential for ID theft detection [3, 4].

Furthermore, sophisticated fraud qualities that a person might miss can be found in ML systems. Identifying patterns is one way to considerably enhance the accuracy of fraud detection, which is an important step in the prevention of identity theft. For instance, if a person's behavior patterns are kept in a database. In this manner, everything that happens in the account is continuously compared to the historical behavior patterns that a certain user has documented. Fraud may be detected if this conduct significantly deviates from the norm. Every new transaction feeds into the model's behavioral fraud analytics process, improving its training [5]. Identifying identity theft is considered an anomaly detection challenge. Several cutting-edge unsupervised machine learning algorithms [6], such as LOF, PCA, one-class SVM, and Isolation Forest, help find strange patterns in a user's behavior to spot actions that aren't allowed, as shown in Figure 1. These algorithms are used to identify deviations from the mainstream by creating dense clusters of deviant behavior data points, which are distinct from normal behavior clusters. Unsupervised and supervised machine learning methods are also viable options for combating fraudulent models [8, 9].

Figure 1: Different Models for Finding Identity Theft

While traditional classification methods are used in the first case, anomaly detection strategies are an alternative in the second scenario. Furthermore, using neural networks is efficient, yet it requires a large amount of data. The banking and retail industries are vulnerable and will continue to see a high volume of fraud cases as long as card-not-present transactions remain prevalent in today's world. Criminal attacks on user data, particularly credit card and false accounts fraud, identity theft, fraudulent transactions, fraudulent emails, and document forgeries, cause many data breaches [10]. Organizations are benefiting from new and improved techniques for fraud detection and prevention that are based on machine learning algorithms because of their enhanced speed, efficiency, and real-time work. This is in contrast to other approaches, which are based on rule-based algorithms for fraud detection, which are becoming increasingly obsolete.

This article aims to investigate fraudulent activities involving credit cards using machine learning techniques to develop a fraud detection system, utilizing innovative and potent methods from rule-based systems.

II. RELATED WORKS

In cases of high data imbalance, class accuracy holds greater significance than overall accuracy; a majority of prior studies have not assessed their models based on categorical accuracy.

In [11], the authors presented a unique deep ensemble learning-based prediction framework (DEAL) for real-time data stream fraud detection. The suggested approach is resilient to latent transaction patterns, like spending habits, and adaptive to data imbalances. Tensors using real data from a big bank are fed into an ensemble of deep learning networks in order to develop a model and predict fraud. It is recommended to use adaptive optimisation in order to reduce the objective function and enhance fraud prediction. Using the scikit-learn, Google TensorFlow, and Keras Deep Learning libraries, all experiments are carried out in Python. The outcomes show that it is more effective than cutting-edge techniques at identifying fraud.

In [12], the authors investigated six ML models for the credit card transaction dataset, which was assembled from Kaggle data. It portrays transactions with doubt as fraudulent and classifies them as belonging to the "high-quality class," whereas genuine transactions are classified as belonging to the "poor class." With 0.172% of fraud instances and real transactions in the relaxations, the dataset is somewhat unbalanced. Python is used to carry out the task. The approaches' presentation is ranked according to the confusion matrix. The results clearly indicate guidelines applied to detect credit card fraud quite precisely. The suggested version could be useful for creating a variety of oddities.

In [13], the authors tested ML models with nonlinear and linear statistical modelling using data collected from credit card transactions. To identify fraudulent transactions, supervised fraud systems are created. Afterwards, the procedures for variable construction, feature selection, data cleaning, data exploration, model algorithms, and outcomes were covered. The study examines and contrasts neural networks, boosted trees, random forests, logistic regression, and support vector machines, which are the five different supervised models. For this specific data set, the boosted tree model exhibits the best fraud detection performance (FDR = 49.83%). A system for detecting credit card fraud may make use of the final model.

In [14], the authors concentrated on four common fraud scenarios in real-world transactions. A number of machine learning models are used to tackle each scam, and an evaluation is used to determine which approach works best. With the aid of a suitable performance metric, this assessment provides a thorough guide for selecting the best algorithm depending on the type of fraud. The detection of fraudulent activity on credit cards in real time is yet another significant and critical area that our solution addresses. An application programming interface (API) module and machine learning models that have been installed are responsible for performing predictive analytics in order to determine whether or not a specific transaction is authentic or fraudulent. The resultant unique approach efficiently tackles the skewed data distribution. A confidential disclosure agreement states that the financial institution provided the data used in the tests.

In [15], the authors generated a high-performance model to identify instances of credit card fraud. They discovered that the hyperbolic tangent activation function and the logistic activation function both perform well in identifying credit card fraud. In the three-hidden layer model, the logistic activation function performs better with 10 nodes (82% sensitivity) and 100 nodes (83% sensitivity), respectively. On the other hand, the hyperbolic tangent activation function works best with 1000 nodes; for 1, 2, and 3 hidden layer counts, its sensitivity is 82%. This research will provide assistance in selecting the appropriate deep learning model to yield the greatest outcomes at the lowest possible cost.

In [16], the authors proposed a framework for the detection of fraudulent credit card transactions. The framework aims to lessen the restrictions that the technologies currently in use pose. In order to evaluate the use of hidden Markov models, a variety of frameworks were proposed for this project. With the help of profiling methods, more tailored Hidden Markov. Models can be constructed and applied to a trainset that is unique to a group of cardholders with comparable payment patterns. To determine the relationship between various transaction classes, clustering algorithms were applied. Testing two different clustering algorithms yielded the most effective clustering algorithm. Additionally, many hidden Markov models were constructed with varying test data conditions.

III. PROPOSED MODEL

In order to determine accuracy, performance is measured, and accuracy is determined based on prediction. Furthermore, a credit card fraud detection model is constructed using classification methods ANN, RF, and DT. After comparing the four experiment-used algorithms, we concluded that SMOTE predicted more accurately than the systems created with the help of the RF, DT, and ANN algorithms.

Dataset description

The dataset includes credit card transactions from European cardholders that took place over two days in September 2013. With a few fraudulent transactions interspersed among numerous records of legitimate transactions, the dataset could be more balanced. Of all transactions, the positive class (frauds) accounts for 0.172% (492 frauds out of 284,807 transactions). The principal components derived with PCA are features V1, V2,... V28; "time" and "amount" are the only features that have not experienced PCA transformation. The seconds that pass between each transaction and the dataset's initial transaction are contained in the feature "Time." The target variable is the 'Class' feature, which has a value of 1 in the event of fraud and 0 otherwise.

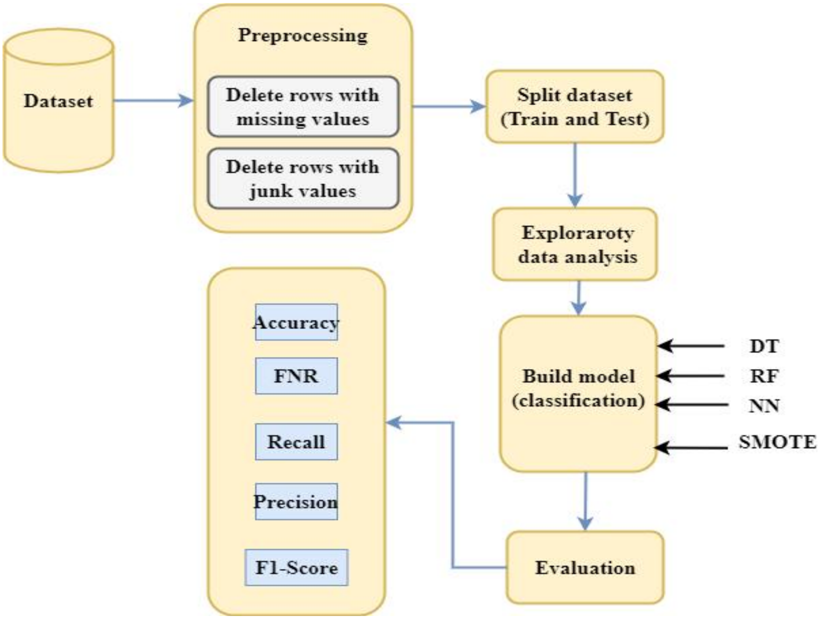
Dataset
Preprocessing
Delete rows with missing values
Delete rows with junk values
Split dataset (Train and Test)
Exploraroty data analysis
Build model (classification)
DT
RF
NN
SMOTE
Evaluation
Accuracy
FNR
Recall
Precision
F1-Score

Figure 2. Process flow of two-stage pipeline of the system

Data Pre-processing:

During this stage, we clean the dataset in order to make it suitable for use in the training of our model. This stage usually entails removing the rows with incorrect or missing values.

Split Dataset:

Each statistically feature-rich constructed dataset is divided into two subsets, referred to as training and testing datasets. Due to the low amount of fraudulent transactions (a total of 284315 authentic transactions and 492 fraudulent transactions), the two types of datasets have been divided into 60% training and 40% testing. Out of the entire number of transactions, about 170589 legitimate transactions and 295 fraudulent transactions were separated in the training section, while approximately 113726 genuine transactions and 197 fraudulent transactions were anticipated in the testing section.

Exploratory data analysis:

The primary objective of our research is to obtain a comprehensive understanding of the dataset including credit card transactions and to devise an efficient model for detecting fraudulent transactions. Exploratory data analysis (EDA) was carried out utilizing popular open source tools like NumPy, Pandas, matplotlib, Seaborn, etc. to gain an understanding of the dataset. Seaborn and Matplotlib are two great packages for visualization. To help us understand the dataset better, we have box plots, density plots, bar graphs, histograms, and other visualizations.

Model Building (Classifier model):

Four classification algorithms were employed in this method to determine which methodology best matched the constructed statistical feature of the dataset to highly accurately suspect the fraud scenario. Many matrices are found to predict and compute as a model to detect the fraud transaction after the classification algorithm is applied to the combined dataset. Using a training dataset to build a model and a testing dataset to predict the outcome, the clustering technique is evaluated. The confusion matrix, which is a matrix format of true and false values, is obtained from the anticipated result.

Decision Tree

A decision tree is a widely used machine learning technique for regression and classification, which involves dividing a dataset into subgroups based on multiple attribute values [14]. The algorithm for guided learning uses a decision tree structure, with a root node and additional nodes divided into child nodes using binary or multi-split methods. Each tree uses its own algorithm to split the data until no further splitting is needed, assigning a value to each input variable related to the method used. With an increasing number of splits, a decision tree model's accuracy increases when applied to a given training dataset. It is advised to utilize the cross-validation procedure because this has the potential to overfit the data. A decision tree model is a straightforward tool that identifies the variable and its proportion used to partition data and predict outcomes.

Random Forest:

The controlled classifier is utilized in various stages of training for decision-making bodies, utilizing a combination of learning technology. Increasing the number of trees in the algorithm helped to raise the system's precision by calculating the exactness of the classifier. During training, the Decision Tree structure generates rules for each class using a testing dataset for monitoring. Subsequently, they employed these principles as the parameters for the assessment data, which can differ from the training data collection, as shown in Figure 3.

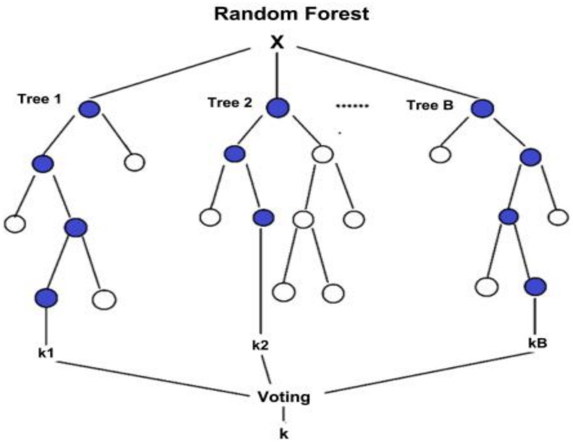

Figure 3: Random forest architecture

The method known as random forest selects a small number of features at the beginning of the process from all of the features offered for every class. A thorough examination of each and every tree that was described during the learning phase is carried out by the test dataset. The target votes were then evaluated by contrasting every prediction target. An advantage of the Random Forest algorithm never materialized when compared to the classification outcomes. Regression, classification, and extraction tasks can also be performed with this technique because the problem model is typically a little error in the Random Forest.

Artificial Neural Network:

ANN is stimulated by the human brain metabolically. The neurons in the human brain are connected to one another in the same way that the nodes in an artificial neural network are connected to one another there. Figure 4 depicts the input, output, and hidden layers that make up an ANN's structure. Y is the output, and the inputs are x_1, x_2,... n. The weights corresponding to inputs $\mathbf{x}_1$–x_n are denoted by w_1–w_n, respectively. This neural network makes use of 25 hidden layers. Our model for detecting credit card fraud uses RELU as its activation function.

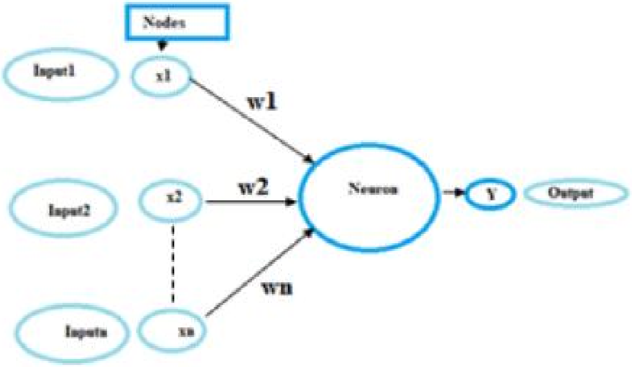

Figure 4: Neural net architecture

We propose a method for oversampling that generates synthetic samples instead of traditional replacement, ensuring disproportionate representation of the minority class. Here is how synthetic samples are produced: Calculate the difference between the sample under examination, including the feature vector and the vector immediately adjacent to it. The difference should be added to the feature vector under examination by multiplying it by a random number between 0 and 1. As a result, a point is chosen at random along the line segment connecting two particular features. This tactic basically forces the zone of decision-making that represents the minority class to become more inclusive. The Step wise Representation for SMOTE is called Algorithm SMOTE.

Algorithm SMOTE (T, N, k)

Input: No. of minority samples of class T

Quantity of SMOTE N%

No. of K nearest neighbour

Output: (N/100)*T minority class samples of T

1. if N< 100%, indiscriminate the class of minority for only as a random percent of them will be SMOTE.

2. if N < 100

3. Next, assign the T samples with a minority class at random.

4. T = (N/100) ∗ T

5. Initialize the value of N as 100

6. end if

7. N as (int)(N/100). (* It is expected that SMOTE is expressed in intrinsic multiples of 100.)

8. k = No of K nearest Neighbour

9. numattrs= No of Attributes

Sample[][]: the Real class of Minority samples in the form of an array

11. newindex: maintains track of the quantity of artificial samples produced, starting at zero

12. Synthetic: An array designed for artificial samples

(∗ Only calculate the k nearest neighbors for each sample of the minority class.)

13. from i → 1 to T

14. Determine k closest neighbors and store the indices in the nn array.

15.occupy (N, i, nnarray) 15.

16. Stop the loop

occupy(N, i, nnarray) (* Function to produce the artificial samples. ∗)

17. While N ≠ 0.

18. Select arbitrarys integer, nn, 1 to k, At this point, pick the one of i's k nearest neighbors.

19. for numattrs to attr → 1.

20. Perform the computation: dif = Sample[narray[nn]][attr] − Sample[i][attr]

21. Calculate: gap = arbitrary number in the interval 0–1.

22. Artificial [new index][attr]=Sample[i][attr] + gap ∗ dif

23. stop the loop

24. new index++;

25. N = N - 1.

26. Stop while loop

27. return the occupy

IV. RESULTS AND DISCUSSION

We have developed and implemented a model using Python platform training four models to identify fraudulent transactions using online credit cards. A number of machine learning methods are examined in relation to the performance metrics found in the dataset for credit card fraud detection. In addition, an oversampling strategy employing different training and testing split ratios was used. The implementation primarily consists of five main machine learning algorithms: DT, RF, NN, and SMOTE. Confusion Matrix is computed using the following parameters: The term "True Positive" (TP) refers to the prediction that a fraudulent transaction is actually taken place. The True Negative (TN) is a prediction that a typical transaction will be performed as expected. The term "false positive" (FP) refers to the situation in which a legitimate transaction is projected to be a fraudulent transaction. A fraudulent transaction is anticipated to be a legitimate transaction, which is referred to as a false negative (FN).

Performance evaluation:

When it came to the grouping, we utilized both of the classifiers. Based on the likelihood of emotional phrase frequency, which was predefined in [6], real-time data labels have been assigned. This allows the classifiers to be

ISSN: 2366-1313

evaluated. It conveys their performance metrics-based success. For evaluating classification efficiency, the f1-score, accuracy, and recall are employed as techniques.

$$\text{Accuracy (Acc)} = \frac{\text{Correctly predicted observations}}{\text{Total number of observations}} \qquad (1)$$

$$\text{Recall} = \frac{\text{Correctly predicted positive observations}}{\text{Total positive observations}} \qquad (2)$$

$$\text{F1-score} = \frac{2}{\dfrac{1}{\text{Precision}} + \dfrac{1}{\text{Recall}}}$$

$$\text{Precision} = \frac{\text{True positive observatoions}}{\text{Actual Result}} \qquad (3)$$

$$\text{Precision} = \frac{\text{True positive observatoions}}{\text{Actual Result}} \qquad (4)$$

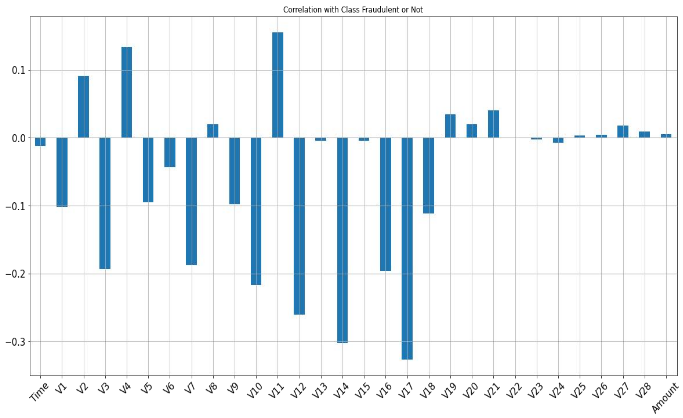

Figure 5: Correlation with Class Fraudulent or Not

Figure 5 shows the correlation with fraudulent or not with its significance values.

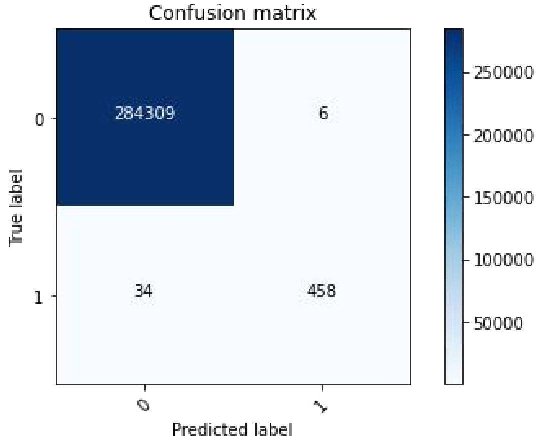

Figure 6: Confusion matrix of Decision tree

Figure 6 displays unfortunately, the model only finds 78% of fraudulent transactions, while 6 normal transactions are wrongly thought to be fraudulent. 33 fraudulent transactions are therefore missed (False Negatives).

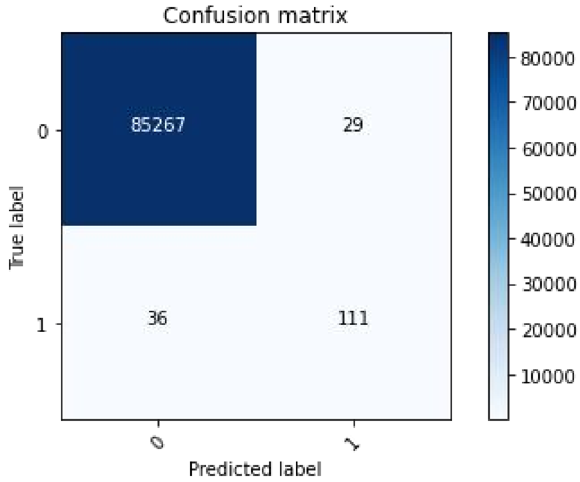

Figure 7: Confusion matrix of Decision Tree

Figure 7 demonstrates the performance of the Decision Tree Structure is below the one using Random Forest. Let's check the performance indicators.

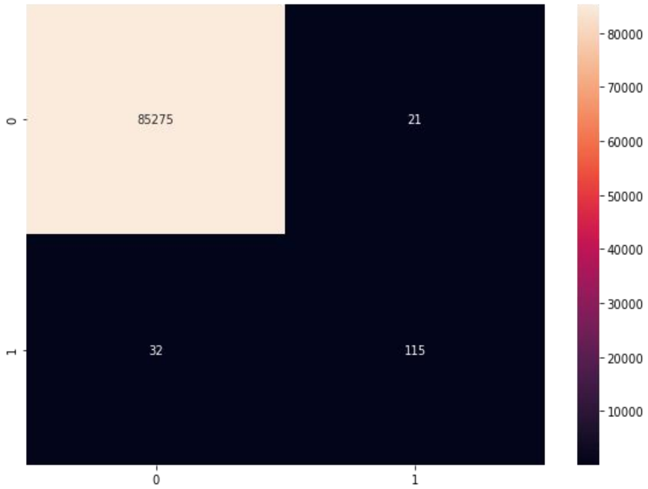

Figure 8: Confusion matrix of neural network

Figure 8 demonstrates that 115 fraudulent transactions are identified by the model as fraudulent; nevertheless, 32 fraudulent transactions are not recognized (false negative), which is still a problem. Given the potentially disastrous effects of fraudulent transactions, our goal must be to identify as many of them as we can. The model identifies 21 regular transactions as possibly fraudulent. These are erroneous positive results. This is a very small amount.

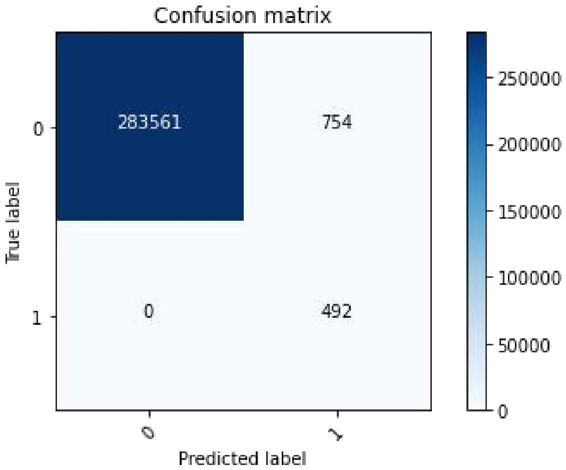

Figure 9: Confusion matrix of Oversampling technique using SMOTE

Figure 9 demonstrates the absence of false negatives. On the whole dataset, the model can identify every fraudulent transaction. Take note of the low amount of False Positives—this indicates that the fraud department will need to verify far fewer transactions (that are actually valid).

Table 1. Represents the accuracy, recall and precision and F1-score

Class	Accuracy	FNR	Recall	Precision	F1 score
RF	0.988	0.069	0.930	0.988	0.957
DT	0.987	0.073	0.926	0.940	0.933
Plain NN	0.989	0.213	0.788	0.854	0.821
Under sample	0.991	0.033	0.967	0.046	0.089
Over sample	0.997	0.000	1.000	0.378	0.566

Figure 10 shows the results demonstrating that the oversampling strategy, which makes use of SMOTE, yields a higher level of accuracy than the unbalanced and under-sampled methods. A normalized and oversampled dataset combined with a combined statistical feature yields 100% recall, 0.9999 accuracy, and precision in the F1 measurement. The SMOTE technique is a method that generates a new vector between two existing data points. By using this method, the amount of fraudulent transactions can be greatly increased.

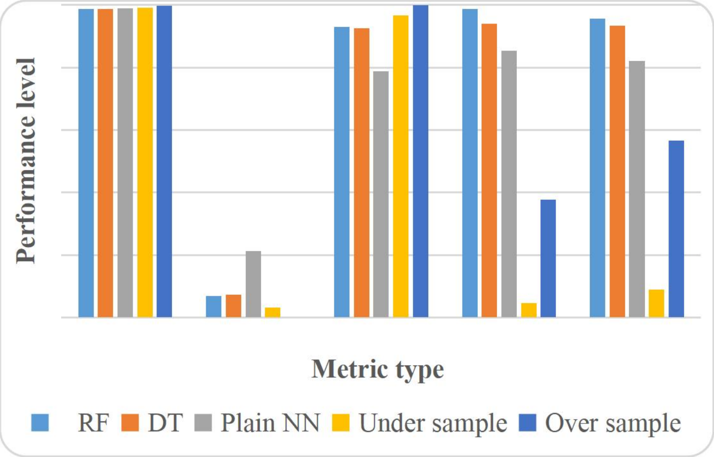

Figure 10: Performance comparison of SMOTE model with DT, RF and NN models

CONCLUSIONS

The confusion matrix produced by each classifier is formulated using various datasets, calculating accuracy, precision, recall, F1-measure, FNR, and execution time. An unbalanced and under-sampled dataset yields less accuracy than an over-sampled dataset. When paired with a normalized and oversampled dataset, a statistical feature yields 100% recall, 0.997 accuracy, precision, F1 measurement, and the right execution time. The synthetic minority oversampling method, or SMOTE, is used to oversample the underrepresented class in order to obtain the best results. By using this method, the model can identify 100% of all fraudulent transactions in the test set that haven't been seen. This completely fulfills the main goal of identifying the great majority of abnormal transactions. Future research can confirm the consistency of the data-point approach in addressing imbalanced credit card fraud datasets by performing cross-validation or comparison across several datasets. More research can look into developing and implementing a real-time system that can identify fraud as soon as the transaction takes place.

REFERENCES

[1] Singh, Soni & Ramkumar, K & Kukkar, Ashima. (2021). Machine Learning Techniques and Implementation of Different ML Algorithms. 1-6. 10.1109/GCAT52182.2021.9586806.

[2] Tae, Chung & Hung, Phan. (2019). Comparing ML Algorithms on Financial Fraud Detection. DSIT 2019: Proceedings of the 2019 2nd International Conference on Data Science and Information Technology. 25-29. 10.1145/3352411.3352416.

[3] Agarwal, Vanshita. (2021). Identity Theft Detection Using Machine Learning. International Journal for Research in Applied Science and Engineering Technology. 9. 1943-1946. 10.22214/ijraset.2021.37696.

[4] Roth, Christian & Nitschke, Mirja & Hutzler, Christian & Koller, Maximilian & Küffner, Rolf & Roßberger, Marc & Kesdogan, Dogan. (2019). My Smartwatch Is Mine – Machine Learning Based Theft Detection of Smartwatches. 10.1007/978-3-030-35055-0_11.

[5] Kumar, Akshi & Anand, Kartik & Jha, Simran & Gupta, Jayansh. (2021). Online Credit Card Fraud Analytics Using Machine Learning Techniques. 10.1007/978-981-15-5616-6_8.

[6] Gain, Ayan. (2021). A Survey on Machine Learning Algorithms. International Journal for Research in Applied

Science and Engineering Technology. 9. 357-366. 10.22214/ijraset.2021.37969.

[7] Pan, Feng & Wang, Weinong. (2006). Anomaly detection based-on the regularity of normal behaviors. 6 pp. - 1046. 10.1109/ISSCAA.2006.1627547.

[8] Alloghani, Mohamed & Al-Jumeily Obe, Dhiya & Mustafina, Jamila & Hussain, Abir & Aljaaf, Ahmed. (2020). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. 10.1007/978-3-030-22475-2_1.

[9] Aissaoui, Ouafae & MADANI, Yasser & Oughdir, Lahcen & EL ALLIOUI, Youssouf. (2019). Combining supervised and unsupervised machine learning algorithms to predict the learners' learning styles. Procedia Computer Science. 148. 87-96. 10.1016/j.procs.2019.01.012.

[10] Parmar, Yogeshvar & Jahankhani, Hamid. (2021). Utilising Machine Learning Against Email Phishing to Detect Malicious Emails. 10.1007/978-3-030-88040-8_3.

[11] Arya, Monika & Sastry, Hanumat. (2020). DEAL – 'Deep Ensemble Algorithm' Framework for Credit Card Fraud Detection in Real-Time Data Stream with Google TensorFlow. Smart Science. 8. 71-83. 10.1080/23080477.2020.1783491.

[12] Parmar, Jasmin & Patel, Achyut & Savsani, Mayur. (2020). Credit Card Fraud Detection Framework - A Machine Learning Perspective. International Journal of Scientific Research in Science and Technology. 431-435. 10.32628/IJSRST207671.

[13] Gao, Jiaxin & Zhou, Zirui & Ai, Jiangshan & Xia, Bingxin & Coggeshall, Stephen. (2019). Predicting Credit Card Transaction Fraud Using Machine Learning Algorithms. Journal of Intelligent Learning Systems and Applications. 11. 33-63. 10.4236/jilsa.2019.113003.

[14] Thennakoon, Anuruddha & Bhagyani, Chee & Premadasa, Sasitha & Mihiranga, Shalitha & Kuruwitaarachchi, Nuwan. (2019). Real-time Credit Card Fraud Detection Using Machine Learning. 10.1109/CONFLUENCE.2019.8776942.

[15] Ramiah Pillai, Thulasyammal & Hashem, Ibrahim & Brohi, Sarfraz & Kaur, Sukhminder & Marjani, Mohsen. (2018). Credit Card Fraud Detection Using Deep Learning Technique. 1-6. 10.1109/ICACCAF.2018.8776797.

[16] Chetcuti, Tanya & Dingli, Alexiei. (2021). Using Hidden Markov Models in Credit Card Transaction Fraud Detection.