Synthetic Minority Oversampling and Smote Regularised Deep Autoencoders Neural Network Techniques for Fraud Prediction in Financial Payment Services

J. Anita Smiles, A. Sasi Kumar

***Abstract:** Frauds in Financial Payment Services are the most prevalent form of cybercrime. The increased growth in e-commerce and mobile payments in recent years is behind the rising incidence of fraud in financial payment services. According to "McKinsey, fraud losses throughout the world could be close to $44 billion by 2025." Every year, fraudulent card transactions causes billions of US Dollar of loss. To reduce these losses, designing effective fraud detection algorithms is essential, which depend on sophisticated machine learning methods to help investigators in fraud. For banks and financial institutions, therefore, fraud detection systems have gained excellent significance. Though the fake transactions are very low when compared to genuine transaction, care must be taken to predict it so that the financial institutions can maintain the customer integrity. As fraud is unlikely to occur compared to normal operations, we have the class imbalance problem. We applied Synthetic Minority Oversampling TEchnique (SMOTE) and the Ensemble of sampling methods(Balanced Random Forest Classifier, Balanced Bagging Classifier, Easy Ensemble Classifier, RUS Boost) to Ensemble machine learning algorithms Performance assessment using sensitivity, specificity, precision, ROC area. The purpose of this article is to analyze different predictive models to see how precise they are to detect whether a transaction is a standard payment or a fraud. Instead of misclassifying a real transaction as fraud, this model seeks to improve detection of fraud. We noted that the technique of Ensemble learning using Maximum voting detects the fraud better than other classifiers. Decision Tree Classifier, Logistic Regression, Balanced Bagging classifier is combined and the proposed algorithm is OptimizedEnsembleFD Algorithm. The sample size is increased and deep learning is applied .It is found that the proposed system Smote Regularised Deep Autoencoders (SRD Autoencoders) neural network performs better with good recall and accuracy for this large dataset.*

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

J. Anita Smiles*, Ph.D Research Scholar, Department of Information Technology, School of Computing Sciences, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, India. E-mail : anitasmiles78@gmail.com

Dr. A. Sasi Kumar, Professor, Department of Information Technology, School of Computing Sciences, Vels Institute of Science, Technology and Advanced Studies, Pallavaram, Chennai, India. E-mail : askmca@yahoo.com

***Keywords:** Fraud detection, Class imbalance, Ensemble classifiers, Ensemble of samples, Synthetic Minority Oversampling TEchnique (SMOTE), Smote Regularised Deep Autoencoders(SRD Autoencoders).*

I. INTRODUCTION

The traditional method of detecting fraud in financial payment services is effective in detecting anomalies that are consistent with known patterns and can not detect fraud that follows fresh or unknown patterns.This is an incentive for criminals to develop ever more advanced, innovative techniques to circumnavigate the rules, and to achieve this, they themselves use new technologies. Rapid advances in the method of machine learning are the solution that helps banks and financial institutions automate the assessment of behavioral patterns of their clients for any indications of abnormality, enabling them to define and flag fraudulent activity in real time. This enables models to adapt over time to uncover patterns that were earlier unknown or to define fresh tactics that fraudsters might use.

The increased accuracy of machine learning provides financial firms with the ability of predicting the number of false prediction in which transactions are flagged incorrectly as fraudulent and declining, and false negatives in which genuine incidences of fraud are missed [1]. Overall, companies can mitigate financial losses, protect their reputations, maintain public trust, and enhance customer experience. Prevention of fraud, attempts to prevent fraudulent transactions at source, and detection helps identify and alert the client as quickly as it is recognized. Detection must therefore always be carried out as no one can predict when there may be a violation of the security provided by techniques of fraud prevention [2],[3]. The quicker a system of fraud detection does, the better.

The count of genuine instances is much more than the unusual instances in fraud detection information. This contributes to the issue of "class imbalance." The conventional machine learning methods expect a balanced class distribution to obtain more precise results [4]. In the last few years, many alternatives have been indicated to solve the issue of learning from imbalanced information sets. The class imbalance issue Data-level, algorithm-level, and ensemble solutions are addressed in three levels. Data-level solutions apply resampling to decrease the adverse impact of class imbalance as a pre-processing phase. Solutions at the algorithm level are aimed at developing fresh algorithms or modifying current ones. In this paper, two types of class balancing methods are used SMOTE and ensemble of samples for balancing the majority and minority classes.

II. RELATED WORK

A. Background

Financial institutions are reluctant to provide confidential data. The lack of information available for studies is the largest study problem. They are also imbalanced. These enormous information is hard to analyze and train as stated in[8]. Many institutes and people have suffered losses across the globe ranging from hundreds to millions of dollars. As the methods of detection and avoidance improved, so did the fraudsters. Extensive research has begun to curb such fraudsters.

The first issue needed to be resolved was the problem of imbalanced learning. Imbalanced data occurs because a data class in the dataset is rare. Although there is a huge problem with credit card fraud, the fraudsters ' ratio to genuine users is very low. This leads to data being imbalanced.

There are many ways in which imbalanced information can be handled. According to[6], when conventional machine learning methods are applied to imbalanced information, the laws of induction defining the majority ideas are often stronger than those of the notion of minority. Class imbalance is studied on a large scale sparse data in a distributed environment, according to[9]. It is regarded here as a issue of cost-sensitive teaching. Sampling methods are also used in[10], where it is shown that the proper preparation of the data set to be analyzed is the main step for correct calculations and accurate predictions. Oversampling methods such as the over-sampling technique of the Synthetic Minority Oversampling TEchnique (SMOTE).

There is a lot of studies going on in detecting fraud. Naïve Bayes, Support Vector machine in[12] are better to identify accuracy and speed of fraud in[3], survey Account Signature for real-time detection of fraud. According to[8], one of the best techniques is to learn ensemble due to their outstanding predictive performance on real-life problems. When building a fraud detection model, it is crucial to apply selection of characteristics and extraction of transactional data characteristics. The writers use cost-sensitive model allocation to boost savings by suggesting a fresh measure comparing a technique's economic cost

III. METHODOLOGY

A. Data Description

Kaggle presents as an approach to such a issue a synthetic dataset produced using the simulator called PaySim. PaySim uses aggregated data from the private dataset to create a synthetic dataset that resembles normal transaction operation and injects malicious behavior to evaluate the performance of fraud detection methods later on. Paysim simulates mobile money transaction based on a sample of actual transactions from one month of economic records from a mobile money service in an African country. The initial logs were given by a multinational company, the mobile financial service provider presently operating throughout the globe in more than 14 nations. This synthetic dataset is ¼ of the initial dataset and is produced for Kaggle only.

Table-1: Dataset Features and Description

Attributes	Description
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	Type of transaction: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.
Amount	Amount of the transaction in local currency
NameOrig	Customer who initiated the transaction
OldbalanceOrig	Initial balance before the transaction
NewbalanceOrig	New balance after the transaction
NameDest	Customer who is the recipient of the transaction
OldbalanceDest	Initial balance of recipient before the transaction
NewbalanceDest	New balance of recipient after the transaction
isFraud	Determines if transaction is fraudulent (encoded as 1) or valid (encoded as 0)
isFlaggedFraud	Determines if transaction is flagged as fraudulent (encoded as 1) or not flagged at all (encoded as 0). An observation is flagged if the transaction is fraudulent and it involved a transfer of over 200,000 in the local currency

B. Data Preprocessing

Exploratory data analysis is performed and it is discovered that out of the five types of transactions, fraud occurs only in two of them 'TRANSFER' where money is sent to a customer / fraudster and 'CASH_OUT' where money is sent to a merchant who pays the customer / fraudster in cash. Remarkably, the number of fraudulent TRANSFERs almost equals the number of fraudulent CASH_OUTs . These observations appear, at first, to bear out the description provided on Kaggle for the modus operandi of fraudulent transactions in this dataset, namely, fraud is committed by first transferring out funds to another account which subsequently cashes it out.

The next phase of the assessment is to figure out the sort in which the fraud transactions occur after balancing the skewed information. It is noted from the dataset that fraud transactions occur only for the CASH-OUT and TRANSFER form. All balancing operations involve mistakes. So we need to analyze whether the mistakes in Genuine and Fraud transactions differ. This must be resolved by including two fresh errorBalanceOrg and errorBalanceDest characteristics. According to the Kaggle dataset overview, the agents ' fraudulent behavior seeks to profit by taking co-operative action.After balancing the skewed data the next step of analysis is to find out the type in which the fraud transactions happens. From the dataset, it is observed that the transactions of fraud happens only for type CASH-OUT and TRANSFER

All transactions which have balance contain errors. So we have to analyse whether there is any difference between the errors in Genuine and Fraud transactions. This need to be solved by including two new features errorBalanceOrg and errorBalanceDest. The frequency at which fraudulent transactions occur appears to alter little over time.

Finally, to encode the categorical values as numbers, we need to conduct 1 hot encoding technique. Before dividing the information into training and testing information, normalization is performed.

C. Proposed System

We first extract the information sets as needed in the process flow (Fig.1).As there is a huge difference between the number of genuine transactions and Fraud transactions Resampling technique is used to balance the classes. SMOTE is used to balance the class here. The Performance is measured using all the existing ensemble classifiers such as RandomForest,AdaBooster,XGBoost,LightGBM,Bagging,GradientBoosting.Among all the existing classifiers RandomForestClassifier works better. The proposed system is developing an Ensemble Learning Algorithm by combining two linear algorithms Decision Tree,Logistic Regression,with the BalancedBagging ensemble (OptimizedEnsembleFD) and the voting classifier is used to find the best prediction .When compared to RandomForest classifier the training time , recall ,f1-score and speed of Proposed System OptimizedEnsembleFD is best. The next step is to create a Deep Neural Network and compare the accuracy to our best classifier. Sample size is increased and deep learning techniques autoencoders is applied. Our SRDAutoencoder(Smote Regularised Deep Autoencoder) uses 4 Desnse (fully connected) layers with 8, 4, 4 and 14 neurons respectively. The first two layers are used for our encoder, the last two go for the decoder. We will train our model for 50 epochs with a batch size of 32 samples. We will use Model Checkpoint to save the best model and TensorBoard for graph visualization. we will use a threshold to separate between fraudulent transactions and legitimate transactions

Autoencoders, which is a deep learning, unsupervised ML algorithm. Our proposed system SRD Autoencoders is a data compression algorithm, which takes the input and going through a compressed representation and gives the reconstructed output. when building the model, 4 fully connected hidden layers were chosen with, [14,7,7,14] number of nodes for each layer. First two for the encoder and last two for the decoder. ROS and RUS data level sampling technique is applied .It is a regularized multilayer deep neural network .Testing set has both normal and fraud transactions in it. From this training method, The model will learn to identify the pattern of the input data. So that we can identify the anomalies of the data. To calculate the error, it uses Mean Squared Error (MSE).

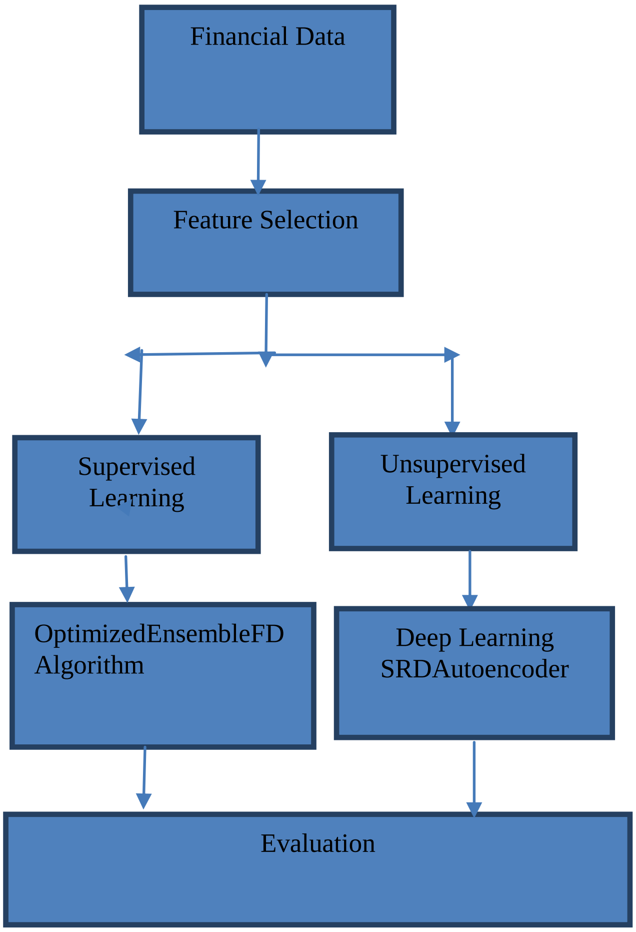

Fig 1: Process Flow of proposed system

IV. EXPERIMENTAL METHODS

A. Resampling Strategies

Some methods are used to overcome the issue of imbalanced data. Data points are created artificially similar to the under-represented class. Undersampling is a technique that randomly samples the dominant class to reduce its size. SMOTE is a method that combines the above two approaches to interpolate the noisy data points at the boundary between outliers and inliers using the KNN algorithm to determine minority classes in the dataset and to learn their features, making the datasets clean and easy to distinguish[13].

We use Ensemble classifiers like EasyEnsemble, RusBoostClassifier,BalancedRandomForest,BalancedBagging, and SMOTE for the resampled information in the suggested scheme. We use altered versions of current classifier algorithms in ensemble resampling approach to create them appropriate for class imbalance issue. By using these two methods, the data fed to a machine learning algorithm is found to be much better balanced and solves the overfitting problem.

B. Ensemble Classifiers

The weak models are properly combined in the Ensemble method to obtain better results and robust models. The three methods for combining weak models are bagging, boosting and stacking.

In Bagging it often feels like homogenous, weak students learn each other in parallel and mix it in a deterministic approach.

In Stacking, heterogeneous weak learners are often viewed, learnt in parallel and are combined to create a prediction based on distinctive weak models, through the practice of a meta-Model. We use an iterative method in Boosting to enhance the easy method of boosting[15]. It focuses on the hard-to-classify patterns. The training subsets are taken randomly (with substitute) from the training set in bagging. Bagging attempts to increase precision by generating an enhanced aggregated classifier, I*, by combining different classifier results into a single prediction [16].

C. Bagging

Random Forest is a ensemble machine learning algorithm which is also known as Bootstrap Aggregation or Bagging. It randomly choosing the subset of feature to create more robust models. Multiple samples of training data are created.The model is constructed for each data sample. For the classification metric prediction the predictions of multiple model are averaged to give a better estimate of the output value.Random Forest is a method where the decision tree is created using suboptimal splits made by randomness. This is the best method as it combines the better estimates of true output underlying value.

$$s_L(.) = \sum_{l=1}^{L} c_l \times w_l(.) \qquad \text{where } c_l\text{'s are coefficients and } w_l\text{'s are weak learners}$$

D. Boosting

It is an ensemble method that tries to produce from a number of weak classifiers a powerful classifier. This is achieved by constructing a model from the training data, then producing a second model to try to correct the first model's mistakes. Models are added up to completely predicting the instruction set or adding a maximum amount of models.

AdaBoost was the first really effective binary classification boost algorithm created. It is the best point of departure for boosting knowledge. Modern techniques of boosting are based on AdaBoost, most notably stochastic machines that boost gradient. With brief decision trees, AdaBoost is used. The efficiency of the tree on each training example is used after the first tree is formed to weigh how much attention each training instance should be given to the next tree being created. More weight is provided to training information that is difficult to predict, while less weight is given to cases that are simple to predict. Models are developed sequentially one by one, each updating the weights on the training cases affecting the learning of the next tree in the series. After all the trees are constructed, projections are created for fresh information, and how precise it was on the training information weighs the efficiency of each tree.

$$s_L(.) = \sum_{l=1}^{L} c_l \times w_l(.) \qquad \text{where } c_l\text{'s are coefficients and } w_l\text{'s are weak learners}$$

E. Autoencoders

Autoencoders are a particular sort of neural architecture network in which the output is identical to the entry. Autoencoders are taught unattended to learn the exceptionally small amount of input information representation. These low-level functions are subsequently deformed to project the information. An autoencoder is a regression task in which the network is asked to predict its input (the identity function, in anderen words). These networks have a close bottleneck of some neurons in the center, which forces them to produce efficient depictions compressing the input into a lower-dimensional code that the decoder can use to replicate the initial input. We will create an autoencoder model in which we only show the model non-fraud cases. The model will try to learn the best representation of non-fraud cases. The same model will be used to generate the representations of fraud cases and we expect them to be different from non-fraud ones.Create a network with one input layer and one output layer having identical dimensions ie. the shape of non-fraud cases. We will use keras package.Create the model architecture by compiling input layer and output layers. Also add the optimizer and loss function, I am using "adadelta" as the optimizer and "mse" as the loss function.

F. Performance Measures

The following performance measures that can offer more insight into the model's precision than traditional classification precision:

- Confusion Matrix: a breakdown of predictions into a table displaying right (diagonal) predictions and kinds of inaccurate predictions (what classes were allocated wrong predictions).
- Precision: a measure of accuracy of the classifier.
- Recall: a measure of completeness of the classifiers
- F1 Score (or F-score): a weighted average of accuracy and recall.

V. COMPARING ACCURACIES AND RESULTS

The experimental findings leading to our conclusion are pro vided in the following section. Table2 summarizes the amou nt of frauds, the number of non-frauds and the imbalance ratio in our datasets.

Table-II: Datasets with Different Imbalance Ratio

Total Transactions	Legitimate	Fraud
6362620	6354407	8213

Only valid transaction involved amounts larger than 10,000,000, however these transactions make up less than 0.01% of the relevant data. When the amounts moved is less than 10,000,000 there doesn't seem to be large difference fraudulent and valid transactions Shown in Fig 2, Table 3 to 8 and Fig 3 to 9 shows the experimental outcomes.

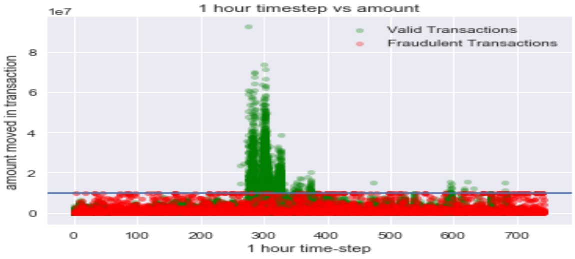

Fig 2: Proportion of transactions where the amount moved is greater than 10 million

Table-III: Performance of SMOTE with Existing ENSEMBLE CLASSIFIERS

Ensemble Classifier with SMOTE	Accuracy	Precision	Recall	Fbetascore	ROC-AUC-Score
RandomForest	0.999937915	0.982134	0.996475	0.993574	0.9982116
Bagging	0.99990182	0.970098	0.996475	0.991086	0.9981935
GradientBoosting	0.998948893	0.732963	0.996475	0.929632	0.997715667
AdaBoost	0.997031488	0.791315	0.996979	0.826791	0.997005246
XGBoost	0.999842623	0.952153	0.995498	0.986516	0.997676474
LightGBM	0.999852729	0.953757	0.996979	0.988024	0.998419923

Table-IV: Performance of Existing Ensemble Sample Techniques

ENSEMLES	Accuracy	Precision	Recall	Fbeta-score	ROC-AUC- Score
BalancedBagging	0.999842623	0.952153	0.995498	0.986516	0.997676474
EasyEnsemble	0.995698835	0.415072	0.99339	0.7769	0.994547954
RUSBoost	0.99929108	0.803985	0.995468	0.950207	0.997385176
BalancedRandomForest	0.99949466	0.860816	0.995751	0.965483	0.997628426

Table–V: Performance of Proposed System-OptimizedEnsembleFD with SMOTE

Confusion Matrix	Genuine	Fraud
Genuine	828616	26
Fraud	8	2473

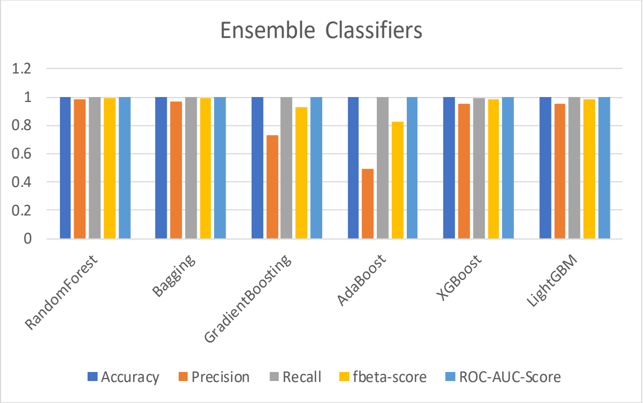

Fig 3. Performance of Ensemble Classifiers with SMOTE

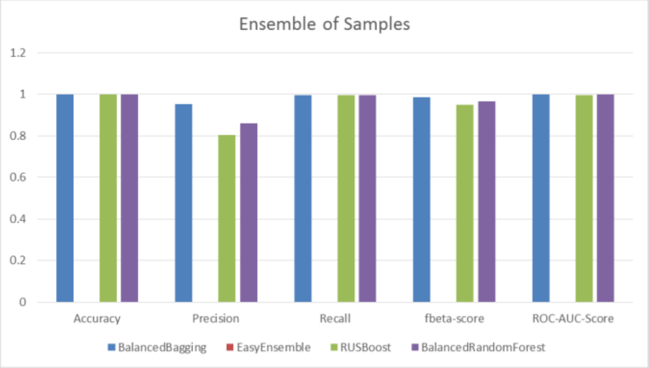

Fig 4. Performance of Ensemble Classifiers with Ensemble of Samples

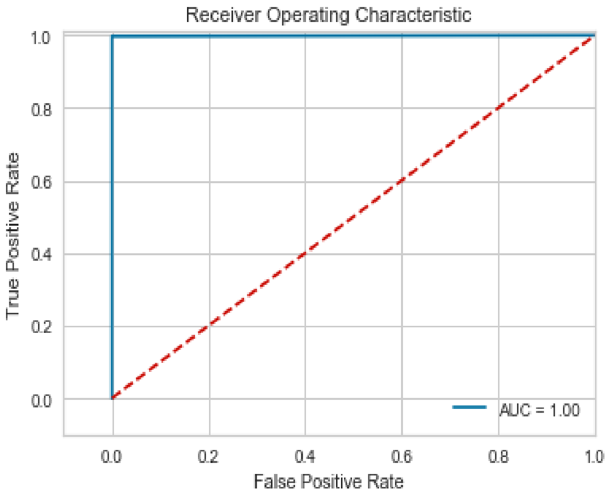

Fig 5. AUC-ROC Curve

Classification Report

Accuracy: 0.9999590914942794
Precision: 0.9895958383353342
Recall: 0.9967754937525192
ROC-AUC-Score: 0.9983720585572993
F1-score: 0.9953312404411175

Feature Importance using OptimizedEnsembleFD Classifier

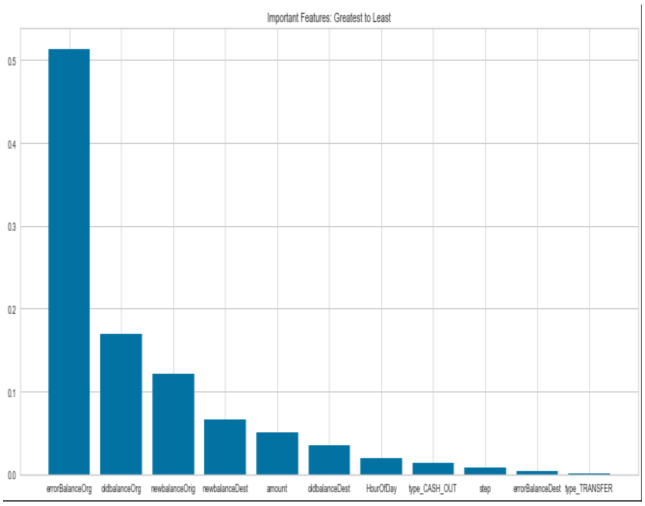

Fig 6. Feature Importance using Ensemble Algorithm

Table VI: Deep Neural Networks in Existing System

	Accuracy	Precision	Recall
SAE	81.83	0.7423	0.722
RBM	92.86	0.8143	0.799

Proposed System- Smote Regularised Deep Autoencoder(SRD Autoencoder)

MSE is basically the reconstruction error, if the model has learned the normal datapoints well, then this error should be less on normal points, and high on fraud points. Create the model architecture by compiling input layer and output layers. Also add the optimizer and loss function, I am using "adadelta" as the optimizer and "mse" as the loss function.

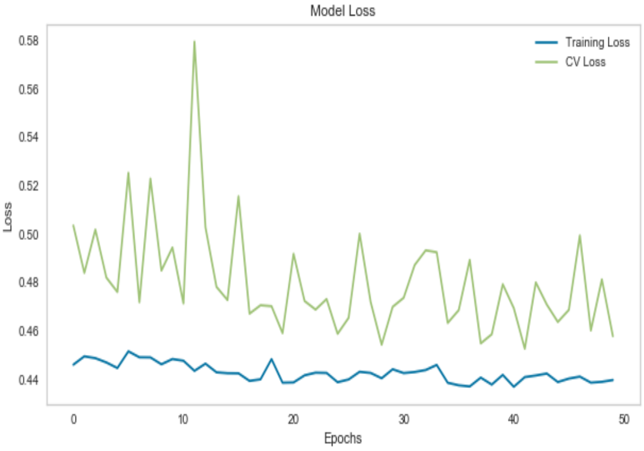

Fig 7. Model Loss

Table VII: Reconstruction Error

	Reconstruction_error							
	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
true_class								
0	828642	0.407654	6.756681	0.023129	0.062857	0.124055	0.317591	2612.2888
1	2481	8.458292	164.49581	0.042106	0.454929	0.881014	1.804497	6936.8604

Table VIII: Classification Report using Proposed SRDAutoencoder Deep Neural Network

Class	**precision**	**recall**	**f1-score**	**support**
0	1.00	0.88	0.94	828642
1	0.02	0.71	0.04	2481
Weighted Average	1.00	0.88	0.94	831123

Area under ROC: 0.7963773872274061

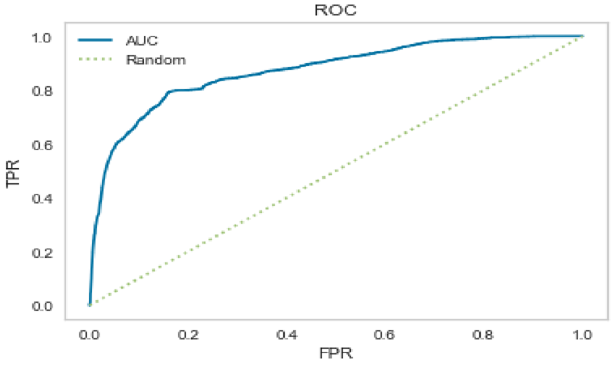

Fig 8. AUC-ROC

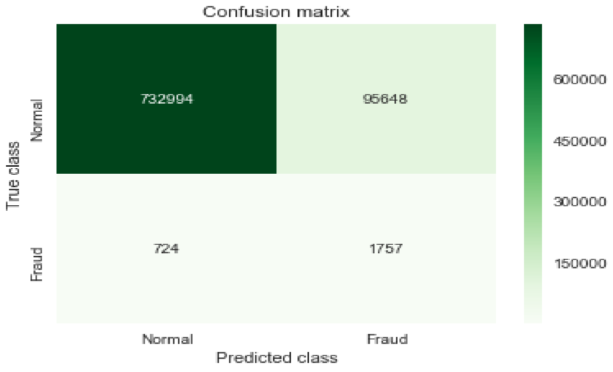

Fig 9. Confusion Matrix

VI. CONCLUSION

SMOTE and set of Ensemble samples are implemented to balance the unbalanceddataset. The assessment of the learning model is based on its accuracy, recall, accuracy, andROC-AUC specificity. After resampling, the findings of all models were superior in overall performance. Overall findings shows OptimizedEnsembleFD classifier is most promising to predict fraud transaction in the dataset. Proposed System SRD Autoencoder performs better for Large dataset. Since the data is highly unbalanced, it cannot be measured only by using accuracy. Precision vs Recall and Area under ROC is chosen for the classification task.

VII. FUTURE WORK

We can enhance the general efficiency by training better models using Deep Neural Networks that extract more abstract characteristics from the data. We need to choose the best model with a decent recall score so that a fraudulent transaction is not to be predicted as a nonfraudulent one, because the result can be very bad for the financial sector an d the bank.Rather than aiming for overall accuracy on the entire dataset, we care more about catching most of the fraud cases (recall), whilst keeping the cost at which this is achieved under control (precision). Finding the optimal minimum and intersect between two cost functions: the cost of fraud and the cost of labor is also the most important work need to be implemented in the future.

REFERENCES

1. Andrea Dal Pozzolo, Olivier Caelen, Reid Johnson and Gianluca Bontempi A.”Calibrating Probability with Undersampling for Unbalanced Classification,” in *In IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2015, pp. 159–166.
2. Agresti, Alan. (2002).” Categorical Data Analysis.” New York:Wiley-Interscience ChengLi.”AGentleIntroductiontoGradientBoosting”
3. Varsha Babar, Roshani Ade, A Novel Approach for Handling Im-balanced Data in Medical Diagnosis using UndersamplingTechnique,Communications on Applied Electronics (CAE), Foundation of Computer Science FCS, New York, USA -2015.
4. B. Adrian, “Detecting and Preventing Fraud with Data Analytics,” Procedia Economics and Finance, vol. 32, no. 15, pp. 1827–1836, 2015.
5. H. He and E. A. Garcia, “Learning from Imbalanced Data,” IEEE Transactions on knowledge and data engineering, vol. 21, no. 9, pp. 1263–1284, 2009.
6. B. Zhu, B. Baesens, and K. L. M. Seppe, “An empirical comparison of techniques for the class imbalance problem in churn prediction,” Information Sciences, vol. 408, pp. 84–99, 2017.
7. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
8. C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the imbalanced class problem,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5476 LNAI, pp. 475–482, 2009.
9. H.Falaki, “AdaBoost Algorithm,” Startrinity. [Online]. Available: http://startrinity.com/VideoRecognition/Resources/Adaboost/boosting algori%0Athm 2.pdf.
10. J. R. Quinlan, “Bagging, boosting, and C4.5,” in Proceedings of the Thirteenth National Conference on Artificial Intelligence, 2006, vol. 5, pp. 725–730.
11. Krishna Modi, Reshma Dayma.” Review On Fraud Detection Methods in Credit Card Transactions” 2017 International Conference on Intelligent Computing and Control (I2C2'17)
12. M. A. Scholar, M. Ali, and P. Fellow, “Investigating the Performance of Smote for Class Imbalanced Learning : A Case Study of Credit Scoring Datasets,” vol. 13, no. 33, pp. 340–353, 2017.
13. M. Hegazy, A. Madian, and M. Ragaie, “Enhanced Fraud Miner: Credit Card Fraud Detection using Clustering Data Mining Techniques,” Egypt. Comput. Sci., no. 03, pp. 72–81, 2016.
14. Jianrong Yao, Jie Zhang, Lu Wang.” Financial Statement Fraud Detection Model Based on Hybrid Data Mining Methods” 2018 International Conference on Artificial Intelligence and Big Data-978-1-5386-6987-7/18/$31.00 ©2018 IEEE
15. Dilip Singh Sisodia, NerellaKeerthana Reddy, Shivangi Bhandari.”Performance Evaluation of Class Balancing Techniques for Credit Card Fraud Detection” IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017)
16. Sahil Dhankhad, Emad A. Mohammed ,”Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study” 978-1-5386-2659-7/18/$31.00 ©2018 IEEE DOI 10.1109/IRI.2018.00025
17. K. Chaudhary, J. Yadav, and B. Mallick, “A review of Fraud Detection Techniques: Credit Card,” Int. J. Comput. Appl., vol. 45, no. 1, pp. 975–8887, 2012.
18. F. N. Ogwueleka, “Data Mining Application in Credit Card Fraud Detection System,” vol. 6, no. 3, pp. 311–322, 2011.
19. G. Rushin, C. Stancil, M. Sun, S. Adams, and P. Beling, “Horse race analysis in credit card fraud - Deep learning, logistic regression, and Gradient Boosted Tree,” 2017 Syst. Inf. Eng. Des. Symp. SIEDS 2017, pp. 117–121, 2017.
20. H. He, W. Zhang, and S. Zhang, “A novel ensemble method for credit scoring: Adaption of different imbalance ratios,” Expert Syst. Appl., vol. 98, pp. 105–117, May 2018.
21. K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, “Credit Card Fraud Detection Using AdaBoost and Majority Voting,” IEEE Access, vol. 6, pp. 14277– 14284, 2018.

22. M. Zareapoor and P. Shamsolmoali, “Application of credit card fraud detection: Based on bagging ensemble classifier,” Procedia Comput. Sci., vol. 48, no. C, pp. 679–686, 2015.
23. Wang, L. and C. Wu, “A Combination of Models for Financial Crisis Prediction: Integrating Probabilistic Neural Network with Back-Propagation based on Adaptive Boosting”. International Journal of Computational Intelligence Systems, 2017. 10(1): p. 507.
24. Wu, L Wang and J Shi, Financial Distress Prediction Based on Support Vector Machine with a Modified Kemel Function: Journal of Intelligent Systems, Journal of Intelligent Systems, Vol 33, No.33, 2015, pp. 177-187.