Credit Card Fraud Detection using SMOTE and Ensemble Methods

Yiquan Xiao[1*], Jinguo Lian[2]

[1]College of Information & Computer Sciences, University of Massachusetts Amherst, USA
[2]Department of Mathematics and Statistics, University of Massachusetts Amherst, USA
*Corresponding Author

Received: 23 July 2021/ Revised: 04 August 2021/ Accepted: 12 August 2021/ Published: 31-08-2021

***Abstract*—** *We focused on the study of using math modeling and machine learning to do big data analysis, therefore to detect Credit card fraud, which is one of the serious issues in real life. In order to detect credit card fraud, after reviewed many recent research, we chose the most popular models among credit card fraud detection, which are Random Forest (RF), and ANN with multi-layers (DNN). We evaluated the accuracy and recall of these models in detecting credit card fraud with or without SMOTE, and found out that there is no significant improvement in the accuracy of these models with or without SMOTE training, but RF with SOMTE has a little bit vantage than others. There is a significant improvement in recall of these three models with SMOTE training. Especially, with SMOTE training, ANN or DNN is of better performance in the recall than RF. Therefore, we combine RF and DNN to generate a hybrid model so that it produces better stability in accuracy and recall. The study discovered that neural network models have greater potential for finding abnormal data in the big data stream. This has important guiding significance for what mathematical model that credit card companies use to monitor the cash flow and remind customers of the possible risk of credit card fraud.*

***Keywords*— *Artificial Neural Network, Credit Card Fraud Detection, Hybrid Model, Random Forest, SMOTE*.**

I. INTRODUCTION

Credit cards are convenient to use and easy to carry. It not only supports off-line payment, but also online payment. With the development of internet technology, more and more people are using credit cards. Nowadays, most people choose to use credit cards for transactions. However, with the growth in the use of credit card transactions, credit card fraud is also on the rise.

To reduce the growing number of credit card frauds, many methods have been developed to detect the fraud. Among them, machine learning models have been proved to be good solutions for credit card fraud detection. There are various machine learning models, either supervised or unsupervised, such as logistic regression, support vector machine (SVM), random forest (RF), k-nearest neighbor, and k-means clustering. Besides these models, Neural networks became popular in recent years, and it was proved to be powerful in many fields, including credit card fraud detection. In 2014, Sitaram patel and Sunita Gond found that the SVM algorithm with user profile instead of only spending profile can improve TP (true positive), TN (true negative) rate, and decreases the FP (false positive) & FN (false negative) rate [7]. In 2017, S. Akila and U. Srinivasulu Reddy analyzed the internal factors that affect the abnormal data found in the credit card transaction and tried to find a way to eliminate these factors. Simulation experiments proved that Non-overlapped Risk based Bagged Ensemble model (NRBE) can improve performances of 5% in terms of BCR and BER, 50% in terms of Recall and 2X to 2.5X times reduced cost [1]. Their research provided an idea for later research, that is, a new method can be used to re-sample existing historical data to generate more efficient training data, thereby improving the accuracy and recall of detecting credit card fraud. In 2019, Devi Meenakshi. B, Janani. B, Gayathri. S, and Mrs. Indira. N discovered that the RF can improve the

accuracy of detecting fraud, even if some data has been missing or has not been scaled well. The RF algorithm will perform better with a larger number of training data, but speed during testing and application will suffer [6]. Simi M.J. evaluated three machine learning supervised algorithms: RF, SVM and ANN, and pointed out their respective pros and cons, and concluded that ANN has the best performance [4]. In 2020, Altyeb Altaher Taha and Sharaf Jameel Malebary explored a new algorithm-an optimized light gradient boosting machine (OLightGBM) to detect fraud in credit card transactions and used F1-score as an indicator to evaluate the quality of an algorithm [12]. In 2021, Asha RB and Suresh Kumar KR confirmed again that ANN machine learning algorithms are of better accuracy than the unsupervised learning algorithms [9].

On the basis of summarizing previous studies, we evaluate the performance in detecting credit card fraud of three models: RF model, ANN model (with 1 hidden layer) and DNN model with or without SMOTE. It turns out that these three models with SMOTE are all of better performance than ones without SMOTE, and eventually we combine the results of RF and DNN models to produce a hybrid model with higher stability.

II. DATA AND REQUIREMENTS

2.1 Data Description

The data set used is the Credit Card Fraud Detection data set from Kaggle. This data set contains credit card transactions in September 2013 in European. There are a total of 284,807 transactions in the data set, but only 492 of them are frauds. Features 'V1', 'V2', …, 'V28' are the principal components obtained with Principal Component Analysis (PCA) in order to protect user privacy. These could be features that are potentially relevant to credit card transactions, such as gender, age, loan annuity, and income. The rest features are 'Time', 'Amount', and 'Class'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the data set. Feature 'Amount' is the transaction Amount. And feature 'Class' is the response variable: 1 means this transaction is a fraud and 0 means it is not. Fig.1, Fig.2, Fig.3, Fig.4 are descriptions of the data.

	Time	V1	V2	V3	V4	V5	V6	V7	V8
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.918649e-15	5.682686e-16	-8.761736e-15	2.811118e-15	-1.552103e-15	2.040130e-15	-1.698953e-15	-1.893285e-16
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01

FIGURE 1: Data Description Part 1

V9	V10	V11	V12	V13	V14	V15	V16	V17
2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
-3.147640e-15	1.772925e-15	9.289524e-16	-1.803266e-15	1.674888e-15	1.475621e-15	3.501098e-15	1.392460e-15	-7.466538e-16
1.098632e+00	1.088850e+00	1.020713e+00	9.992014e-01	9.952742e-01	9.585956e-01	9.153160e-01	8.762529e-01	8.493371e-01
-1.343407e+01	-2.458826e+01	-4.797473e+00	-1.868371e+01	-5.791881e+00	-1.921433e+01	-4.498945e+00	-1.412985e+01	-2.516280e+01
-6.430976e-01	-5.354257e-01	-7.624942e-01	-4.055715e-01	-6.485393e-01	-4.255740e-01	-5.828843e-01	-4.680368e-01	-4.837483e-01
-5.142873e-02	-9.291738e-02	-3.275735e-02	1.400326e-01	-1.356806e-02	5.060132e-02	4.807155e-02	6.641332e-02	-6.567575e-02
5.971390e-01	4.539234e-01	7.395934e-01	6.182380e-01	6.625050e-01	4.931498e-01	6.488208e-01	5.232963e-01	3.996750e-01
1.559499e+01	2.374514e+01	1.201891e+01	7.848392e+00	7.126883e+00	1.052677e+01	8.877742e+00	1.731511e+01	9.253526e+00

FIGURE 2: Data Description Part 2

V18	V19	V20	V21	V22	V23	V24	V25	V26
2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
4.258754e-16	9.019919e-16	5.126845e-16	1.473120e-16	8.042109e-16	5.282512e-16	4.456271e-15	1.426896e-15	1.701640e-15
8.381762e-01	8.140405e-01	7.709250e-01	7.345240e-01	7.257016e-01	6.244603e-01	6.056471e-01	5.212781e-01	4.822270e-01
-9.498746e+00	-7.213527e+00	-5.449772e+01	-3.483038e+01	-1.093314e+01	-4.480774e+01	-2.836627e+00	-1.029540e+01	-2.604551e+00
-4.988498e-01	-4.562989e-01	-2.117214e-01	-2.283949e-01	-5.423504e-01	-1.618463e-01	-3.545861e-01	-3.171451e-01	-3.269839e-01
-3.636312e-03	3.734823e-03	-6.248109e-02	-2.945017e-02	6.781943e-03	-1.119293e-02	4.097606e-02	1.659350e-02	-5.213911e-02
5.008067e-01	4.589494e-01	1.330408e-01	1.863772e-01	5.285536e-01	1.476421e-01	4.395266e-01	3.507156e-01	2.409522e-01
5.041069e+00	5.591971e+00	3.942090e+01	2.720284e+01	1.050309e+01	2.252841e+01	4.584549e+00	7.519589e+00	3.517346e+00

FIGURE 3: Data Description Part 3

V27	V28	Amount	Class
2.848070e+05	2.848070e+05	284807.000000	284807.000000
-3.662252e-16	-1.217809e-16	88.349619	0.001727
4.036325e-01	3.300833e-01	250.120109	0.041527
-2.256568e+01	-1.543008e+01	0.000000	0.000000
-7.083953e-02	-5.295979e-02	5.600000	0.000000
1.342146e-03	1.124383e-02	22.000000	0.000000
9.104512e-02	7.827995e-02	77.165000	0.000000
3.161220e+01	3.384781e+01	25691.160000	1.000000

FIGURE 4: Data Description Part 4

2.2 Software and Package Requirements

Below are software and packages to repeat the work in this paper.

Programming language used: Python

Python Packages and libraries used:

- Numpy:

  numpy is a Python library used to deal with algebra operations.

- Matplotlib:

  matplotlib is a Python library for plot.

- Pandas:

  pandas is a Python library used to perform common statistical operations on data.

- Imbalanced Learn:

  imbalanced-learn is a python package that offers a number of re-sampling techniques for imbalanced data.

- Scikit Learn:

  scikit-learn is a python package that offers various statistical models and machine learning models.

III. METHODS AND MODELS

3.1 Software and Package Requirements

3.1.1 Synthetic Minority Over-sampling Technique

Synthetic Minority Over-sampling Technique (SMOTE) is one of the re-sampling strategies for imbalanced data sets. It deals with the imbalance by over-sampling minority observations. In the credit card fraud detection problem, fraud cases are always much less than normal transactions. The normal over-sampling method takes random draws from the fraud cases and copies those observations to increase the amount of fraud samples. In this way, the model will be trained on a lot of duplicates. SMOTE, on the other hand, uses characteristics of nearest neighbors of fraud cases to create new synthetic fraud cases, and thus avoid duplicating observations. [5]

3.1.2 Decision Tree

Decision Tree is a type of supervised machine learning that can be used on both classification and regression problems. It is a structure that includes root node, leaf node & branch. Each internal node denotes a test on attribute, the outcome of the test denotes each branch and the class label is held by each leaf node. The root node is the topmost node in the tree. [2]

3.1.3 Random Forest

Random Forest (RF) is a model based on Decision Tree (CART). It is like applying Ensemble method to Decision Trees. RF builds multiple decision trees with different samples and initial variables. And the final prediction of RF combines the results of all the trees. [6] Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output does not depend on one decision tree but multiple decision trees. [8] The code to implement RF in this paper references Sharma's [10] and McKinney's [5] articles.

3.1.4 Artificial Neural Network

Artificial Neural Network (ANN) is a machine learning algorithm inspired by biological neural networks in human brains. In ANN, each node represents a perception in the neural network, and nodes are arranged in layers. This paper uses ANN with back propagation. The implementation of ANN references Sun's [11] article.

3.2 Proposed System

3.2.1 Description

Ensemble methods are techniques that create multiple machine learning models and then combine them to produce improved results. The proposed system uses SMOTE as the re-sampling method to deal with imbalanced data. Then, the system combines the result of RF and ANN using Ensemble methods.

3.2.2 Steps and Parameters

Steps of the proposed model:

- Drop *Time* column and scale the *Amount* columns in the data set.(columns*V1* to *V28* are already processed with PCA)
- Divide data set into training and testing
- Apply SMOTE to the training data set
- Define Random Forest
- Define Deep Neural Network
- Use Ensemble Methods to combine RF and DNN above
- Train the proposed model
- Predict the testing data set using trained model

Table 1 below presents some scikit-learn modules used in the above steps:

TABLE 1
SCIKIT LEARN MODULES USED

Purpose	Module
Scaler	StandardScaler
Divide dataset	train_test_split
Apply SMOTE	SMOTE
Define Random Forest	RandomForestClassifier
Define ANN or DNN	MLPClassifier
Ensemble Methods	VotingClassifier

Table2 below presents the parameter values used in the proposed model:

TABLE 2
SOME PARAMETERS USED

Parameter	Value
Random State	0
Hidden layer sizes of DNN	16, 20, 16, 20 (n^{th} number represent number of nodes in n^{th} hidden layer)
Activation Function	logistic function
Solver of DNN	stochastic gradient descent
Maximum number of iterations	500

IV. EVALUATION

4.1 Evaluation Metrics

4.1.1 Confusion Matrix and Accuracy

The following figure Fig.5shows the composition of the confusion matrix:

		True Condition	
		Condition Positive	Condition Negative
Predicted Condition	Predicted Condition Positive	True Positive	False Positive
	Predicted Condition Negative	False Negative	True Negative

FIGURE 5: Confusion Matrix

True Positive (TP): Cases that model correctly predict as fraud

True Negative (TN): Cases that model correctly predict as non-fraud

False Positive (FP): Cases of 'false alarm'. (Model predict it should be fraud, but actually not)

False Negative (FN): Cases of fraud not caught by the model

Accuracy is the fraction of transactions that were correctly classified. It is one of the most powerful and commonly used evaluation metrics.[3] It can be calculated from Confusion Matrix:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad \#(1)$$

4.1.2 Precision and Recall

Precision and Recall are two important evaluation metrics. Precision represents the fraction of actual fraud cases out of all predicted fraud cases. And Recall represents the fraction of predicted fraud cases out of all actual fraud cases. They can be calculated using confusion matrix:

Precision:

$$Precision = TruePositive/(TruePositive + FalsePositive)\#(2)$$

Recall:

$$Recall = TruePositive/(TruePositive + FalseNegative)\#(3)$$

Although both Precision and Recall are equally important for balanced data, Recall is more important than Precision in Credit Card Fraud Detection. This is because False Negative is worse than False Positive in this problem. (False alarms do not cause much financial loss, but undetected fraud can)

4.2 Model Performance

Since Recall is more important than Precision in this problem, we will use Recall and Accuracy as evaluation metrics.

Performance of RF without SMOTE:	Performance of RF with SMOTE:
the Model used is Random Forest without SMOTE The accuracy is 0.9995084442259752 The recall is 0.7619047619047619 The precision is 0.9411764705882353 Confusion matrix: [[85289 7] [35 112]]	the Model used is Random Forest with SMOTE The accuracy is 0.9995201479348805 The recall is 0.8095238095238095 The precision is 0.9015151515151515 Confusion matrix: [[85283 13] [28 119]]
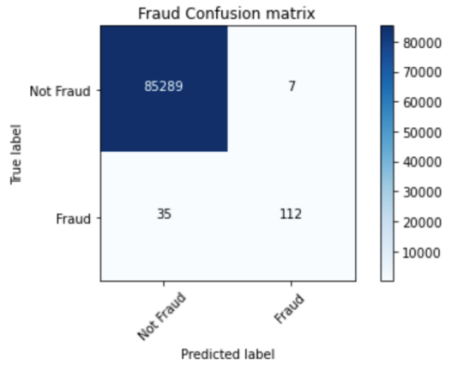	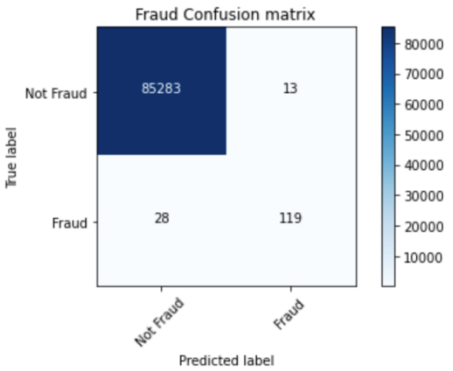
FIGURE 6: Result of RF without SMOTE	**FIGURE 7: Result of RF with SMOTE**
Performance of 1 hidden layer ANN without SMOTE:	**Performance of 1 hidden layer ANN with SMOTE:**
the Model used is 1 layer Artificial Neural Network without SMOTE The accuracy is 0.9988998513628969 The recall is 0.43537414965986393 The precision is 0.8533333333333334 Confusion matrix: [[85285 11] [83 64]]	the Model used is 1 layer Artificial Neural Network with SMOTE The accuracy is 0.9924277003382372 The recall is 0.8639455782312925 The precision is 0.16843501326259946 Confusion matrix: [[84669 627] [20 127]]
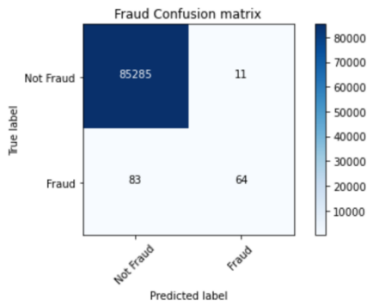	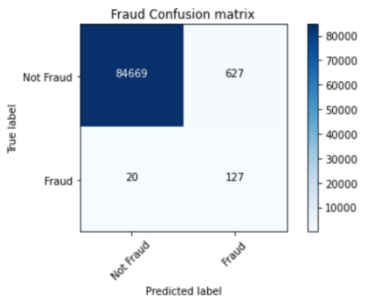
FIGURE 8: Result of 1 hidden layer ANN without SMOTE	**FIGURE 9: Result of 1 hidden layer ANN without SMOTE**

Comparison of models above:

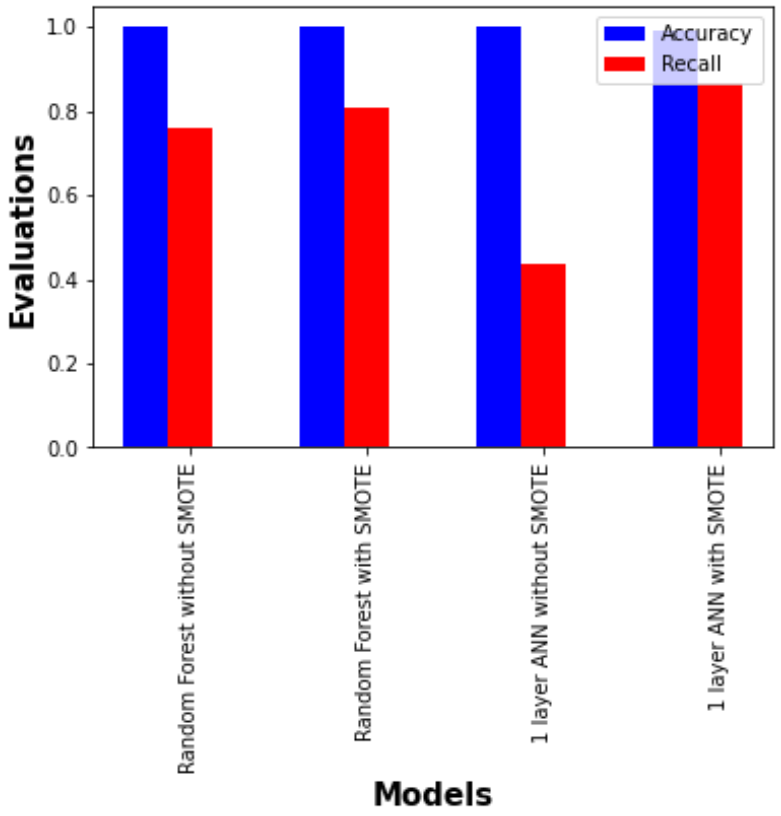

FIGURE 10: ***Comparison of models with and without SMOTE***

We can clearly see that by using the SMOTE method, recall is boosted a lot, especially for ANN.

Performance of Deep Neural Network (DNN) (ANN with multiple hidden layer) with SMOTE:

```
the Model used is Deep Neural Network without SMOTE
The accuracy is 0.9961260723523284
The recall is 0.8639455782312925
The precision is 0.2899543378995434
Confusion matrix:
 [[84985   311]
 [   20   127]]
```

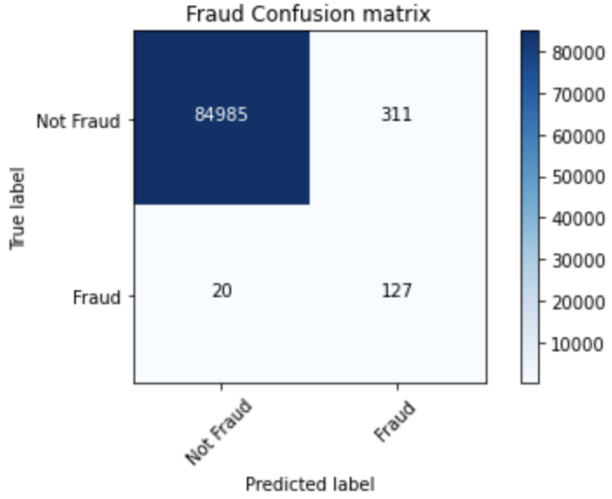

FIGURE 11: Result of deep neural network with SMOTE

Performance of proposed system:

```
the Model used is Proposed Model with SMOTE
The accuracy is 0.9984902215512096
The recall is 0.8571428571428571
The precision is 0.5384615384615384
Confusion matrix:
 [[85188   108]
 [   21   126]]
```

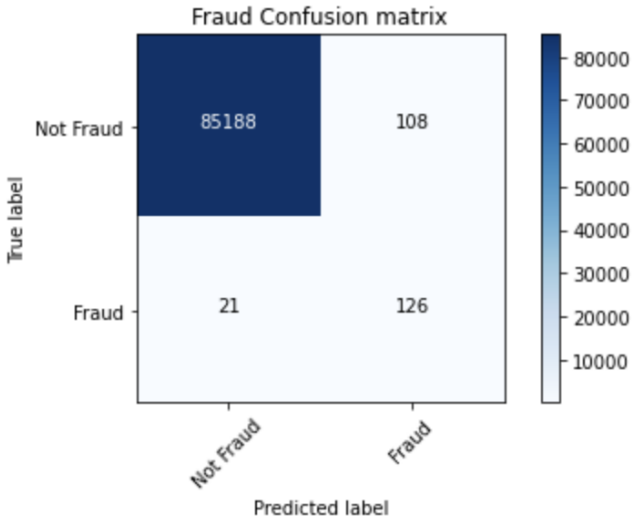

FIGURE 12: Result of proposed system

Comparison of all models:

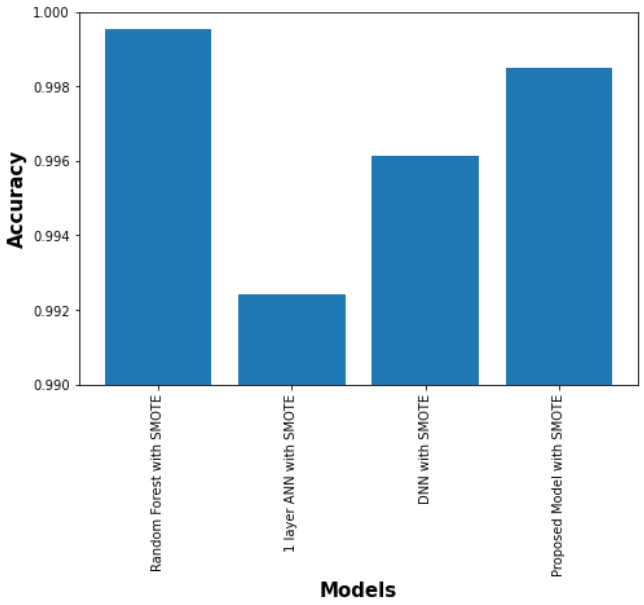

FIGURE 13: Accuracy comparison

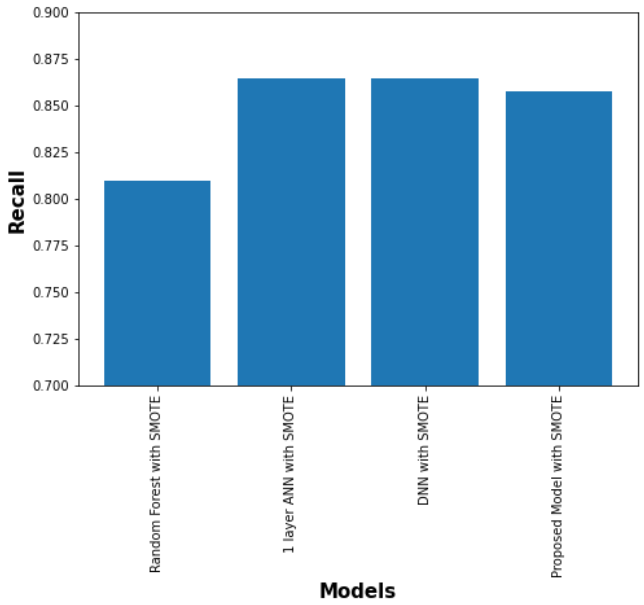

FIGURE 14: Recall comparison

We could see that our proposed model combines the advantages of RF and DNN. For the recall, it performs just a little below ANN, but much better than RF. For accuracy, its performance is really close to RF, but better than ANN. By doing some trade-off, we increase the False Negative by 1 but decrease False Positive by 203 compare to DNN.

Based on the results, it is much better to use SMOTE than not to use SMOTE, especially considering the boost in Recall. And if you are interested in only Recall, DNN with SMOTE is a good model. If you are interested in Accuracy and Precision,

RF is better. And if you want a model with good performance on both Accuracy and Recall, the proposed model is the best choice.

V. CONCLUSION

In this research, we evaluated RF, 1 hidden layer ANN and DNN models with or without SMOTE. After comparison and analysis, we come to the following conclusions:

1) No matter with or without SMOTE training, the accuracy of RF model is of a little bit vantage than ANN and DNN.
2) With SMOTE training, the accuracy of RF, ANN and DNN are all improved, but the recall of ANN and DNN are all of better performance than RF. Especially DNN has better performance than ANN in both accuracy and recall.
3) Based on optimal combination, we generate a hybrid model using RF and DNN with SMOTE training to build a stable performance in both accuracy and recall.

In other words, we proposed a method for credit card fraud detection that is based on SMOTE, Ensemble Methods and some popular existing models. By comparing models with and without SMOTE, we show that applying SMOTE to deal with imbalanced data can increase the model performance. And then, we show that our proposed model is well suited for credit card fraud detection by comparing it to RF and ANN. RF shows its good performance on Accuracy and Precision, while ANN is better on Recall. The proposed model combines the advantages of these two models and provides high recall and high accuracy.

REFERENCES

[1] S. Akila and U. Srinivasulu Reddy. "Credit Card Fraud Detection Using Non-Overlapped Risk Based Bagging Ensemble (NRBE)". In: *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. 2017, pp. 1–4. doi: 10.1109/ICCIC.2017.8524418.

[2] Jyoti R. Gaikwad et al. "Credit Card Fraud Detection using Decision Tree Induction Algorithm". In: *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* vol.4 (Nov. 2014), pp. 66–69.

[3] Yashvi Jain et al. "A Comparative Analysis of Various Credit Card Fraud Detection Techniques". In: *International Journal of Recent Technology and Engineering (IJRTE)* vol.7 (Jan. 2019), pp. 402–407.

[4] Simi M.J. "Credit Card Fraud Detection : A Comparison using Random Forest, SVM and ANN". In: *International Research Journal of Engineering and Technology (IRJET)* vol.06 (Mar. 2019), pp. 225–228.

[5] Trenton McKinney. *Introduction and preparing your data*. July 2019. url:https://trenton3983.github.io/files/projects/2019-07-19_fraud_detection_python/2019-07-19_fraud_detection_ python.html.

[6] Devi Meenakshi.B et al. "CREDIT CARD FRAUD DETECTION USING RANDOM FOREST". In: *International Research Journal of Engineering and Technology (IRJET)* vol.06 (Mar. 2019), pp. 6662–6666.

[7] Sitaram patel and Sunita Gond. "Supervised Machine (SVM) Learning for Credit Card Fraud Detection". In: *International Journal of Engineering Trends and Technology (IJETT)* vol.8 (Feb. 2014), pp. 137–139.

[8] *Random Forest Regression in Python*. May 2020. url:https://www.geeksforgeeks.org/random-forest-regression-in-python/

[9] Asha RB and Suresh Kumar KR. "Credit card fraud detection using artificial neural network". In: Global Transitions Proceedings 2.1 (2021). 1st International Conference on Advances in Information, Computing and Trends in Data Engineering (AICDE - 2020), pp. 35-41. issn: 2666-285X.
doi: https://doi.org/10.1016/j.gltp.2021.01.006. url: https://www.sciencedirect.com/science/ article/pii/S2666285X21000066.

[10] Aman Sharma. *Credit Card Fraud Detection in Python using Scikit Learn.* Oct. 2019. url: https: //medium.com/analytics-vidhya/credit-card-fraud-detection-in-python-using-scikit-learn- f9046a030f50.

[11] Luke Sun. *Credit Card Fraud Detection*. Oct. 2020. url: https://towardsdatascience.com/credit- card-fraud-detection-9bc8db79b956.

[12] AltyebAltaher Taha and Sharaf Jameel Malebary. "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine". In: *IEEE Access* 8 (2020), pp. 25579– 25587. doi: 10.1109/ACCESS.2020.2971354.