Prediction of suspected elder fraud in digital payments

Janani Vijayarajan*
Raahul Kalyaan Jakka*
Varsha Venkata Krishnan*
jvijaya@purdue.edu
jakka@purdue.edu
venka104@purdue.edu
Purdue University
West Lafayette, Indiana, USA

ACM Reference Format:
Janani Vijayarajan, Raahul Kalyaan Jakka, and Varsha Venkata Krishnan. 2022. Prediction of suspected elder fraud in digital payments. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Elders are a particularly appealing target for payment fraudsters, owing to their greater trustworthiness than the younger population. Many of them are unaware of and have limited knowledge on how to use digital payment methods. This pandemic has compelled many people to switch to digital payment systems without properly comprehending the system, making them even more vulnerable to fraud.

It is now exceedingly difficult to identify such instances. When bankers speak with consumers, they listen for clues of such attacks in the dialogue. Many clients are hesitant to confess that they have been harmed, making it extremely rare for them to come forward and report such events to the bank directly. Elders frequently are unaware that they can seek assistance from a bank.

The goal of this study is to detect the possibility of such scams directly from transaction data, without consulting bankers or customers. The method can be used to not just contact customers who may have been victims of an attack, but also to prevent fraudulent transactions from being completed.

Our project is a binary classisfication type of problem which classifies the insatance into fraud and non-fraud.

2 LITERATURE SURVEY

(1) **A Cost-sensitive weighted Random Forest Technique for Credit Card Fraud Detection [1]**

(a) Overview:
This paper primarily addresses the performance drop seen while handling unbalanced banking transaction datasets by traditional methods. The algorithm builds random forest classifiers by assigning costs to each tree while training, to give more importance to the minority class instances. Based on the cost assigned, each tree's predictive ability is assessed and the prediction of the most weighted tree is considered as the model's final prediction.

(b) Method used:
Training method: Dataset is divided into overlapping samples for each tree in the random forest. A C4.5 trainer is used to train the trees. Based on the confusion matrix, the training error is calculated by giving equal importance to both majority and minority class predictions. In case the training error is the same for two different trees, the minority class error is used to break the tie.
Testing method: The test data is passed to the ensemble and the prediction given by the tree which has the highest weight among the trees in the random forest is selected.

(c) Results:
The performance of the proposed method has been compared to that of 'standard random forest' and 'random forest based imbalanced data cleaning and classification'[4]. F-measure, G-mean and AUC values were used to perform the analysis. The proposed algorithm gave a significantly higher value for both the tested datasets for all three measures.

(d) Advantages:
- Handles the dataset imbalance by giving importance to positive examples (minority class)
- Misclassification cost, which is lower on predicting a non fraud as fraud and higher otherwise, has been handled by creation of an ensemble and choosing the model which has low positive error.

(e) Disadvantages:
- If a model has higher overall prediction accuracy but lower minority class accuracy, it will still be selected over a model with lower overall accuracy and higher minority class accuracy.
- Ensemble models perform better when more than one

*All authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

model from the ensemble is considered for prediction.

(f) Improvements:
- The cost function can be modelled as a combination of overall accuracy and positive accuracy to give more weightage to positive sample prediction
- A voting based decision from top k trees in the ensemble or a weighted average of the results produced by the models in the ensemble might distribute the responsibility across multiple trees in the random forest.

(2) **XBNet : An Extremely Boosted Neural Network[5]**

(a) Overview:
This paper discusses a novel architecture which is a combination of tree-based models and neural networks. It also uses a new optimization technique called Boosted Gradient Descent. This helps in improvement in interpretability and performance of the overall model.

(b) Method used:
XBNet inputs raw tabular data and is trained using an optimization technique Boosted Gradient Descent which is initialized with the feature importance of a gradient boosted tree, and it updates the weights of each layer in the neural network in two steps:

(1) Update weights by gradient descent.
(2) Update weights by using feature importance of a gradient boosted tree in every intermediate layer.

Feature importance of an attribute is determined by the result of the difference between the information gain before the split and after the split this attribute in the tree.

At the time of training the model, the data that is fed completes a forward and backward propagation, and the weights of all the layers get updated according to gradient descent once and then instead of going to the next epoch of training it goes through all the layers again and updates its weights again based on the feature importance of the gradient boosted tree that is trained on the layers respectively. To ensure that the contribution of the weights provided by the feature importance and the weights of gradient descent is in the same order the feature importance is scaled down to the same power as that of the weights of the gradient descent algorithm.

(c) Advantages:
Due to the extra feature addition, the models are more reliable when tested on outliers.

(d) Disadvantages:
Extra Preprocessing of the data has to be done before using it for training as the Gradient boosted tree uses information gain for every attribute, during the initialization of weights.

(3) **Credit Card Fraud Prediction and Classification using Deep Neural Network and Ensemble Learning [2]**

(a) Overview:
This paper's aim is to create a model which predicts faulty transactions. This paper has applied four algorithms (Naïve Bayes Classifier Algorithm, Logistic Regression, Decision Trees and Deep Belief Network) to the dataset and then used ensemble learning to get the output. They have compared the result of this output with the ensemble learning using three algorithms (Naïve Bayes Classifier Algorithm (Gaussian), Logistic Regression, Decision Trees) applied to the same dataset. The aim of this paper is to denote the significance of deep belief network algorithms.

(b) Method used
The proposed method includes ensemble learning using four algorithms which are Naïve Bayes Classifier Algorithm (Gaussian), Logistic Regression, Decision Tree and Deep Belief Network. Ensemble learning is used to combine and evaluate various models to produce a merged and better solution. Also, it helps to reduce bias, noise and variance while giving a more legit result. So, this paper combines the four algorithms in order to get an optimal result for the problem.

They have imported Logistic Regression, GaussianNB, Decision Tree Classifier from ScikitLearn Linear Model and then created an instance of the estimator for each model which was then passed for ensemble learning. They have used raw code of DBN to implement it and modified the code and binarized the features to implement DBN perfectly.

(c) Advantages
- Any system that incorporates neural networks needs vast amounts of data for the process to work efficiently and a number of parameters to be set before any training can start. But, this algorithm gives a competing performance even with less data.
- This system can handle enormous data unlike the traditional methods like Support Vector Machines and decision trees.
- Deep belief networks consume lesser time unlike Artificial Neural Networks, which may take up to much longer to train.

(d) Improvements
Could have used weighted voting classifier to improve the classification model's performance by combining the classification results of the single classifier and selecting the group with the highest vote based on the weights given to the single classifiers.

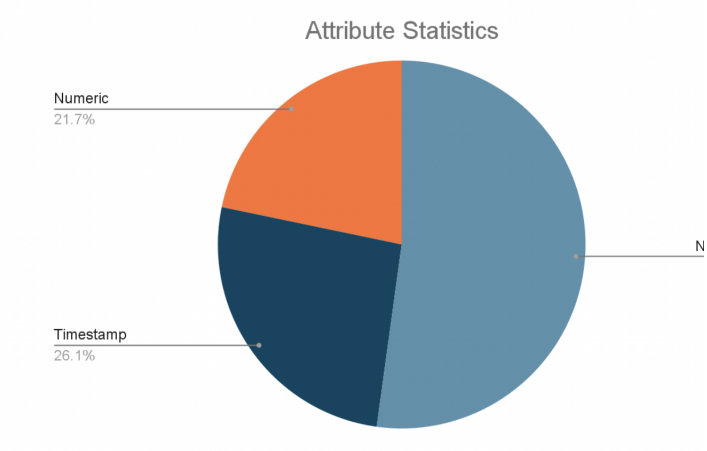

Figure 1: Attribute statistics

Attribute name	*Number of unique values*
CARR_NAME	554
RGN_NAME	19
STATE_PRVNC_TXT	127
ALERT_TRGR_CD	2
DVC_TYPE_TXT	4
AUTHC_PRIM_TYPE_CD	5
AUTHC_SCNDRY_STAT_TXT	3
CUST_STATE	48
ACTN_CD	1
ACTN_INTNL_TXT	1
TRAN_TYPE_CD	1
FRAUD_NONFRAUD	2

Figure 2: Unique value count for each nominal attribute

3 DATASET AND DATA PREPROCESSING:

We used the Wells Fargo - Campus Analytics Challenge 2021 dataset [3] for this project. It is a synthetic dataset which was generated using Conditional GAN to mimic the original dataset.

The Wells Fargo dataset [1] has 14000 rows and 23 attributes. The target attribute is a binary variable denoting if the instance is a fraud or non-fraud.

Attribute statistics:
The attribute statistics are depicted in Figure 1. The number of unique values per nominal attribute is tabulated in Figure 2.
Nominal attributes: 12
Timestamp attributes: 6
Numeric attributes: 5

Number of unique values in each nominal attribute is tabulated in Figure 1.

(1) **Dataset cleaning:**
The attribute '*WF_dvc_age*' denotes the age of the Wells Fargo device, which is a non-negative attribute. All noisy instances with a negative value have been dropped. The attributes *ACTVY_DT*, *TRAN_DT* and *TRAN_TS* had redundant values. So the first two attributes were dropped. Attributes *TRAN_TYPE_CD*, *ACTN_INTNL_TXT* and *ACTN_CD* have only one unique value, so all three attributes were dropped.

(2) **Handling timestamp attributes:**
Timestamps were handled in two different ways as described below.
1. Removal of timestamp attributes
2. Transformation of timestamp attributes
- Missing phone and password update timestamps were filled with account creation time (assuming it was never changed after creation).
- New features were created as follows,
(i)Number of days between change of password and transaction:
Transaction time - password update time
(ii) Number of days between change of phone number and transaction:
Transaction time - phone update time
(iii) Number of days since the customer created an account:
Transaction time - customer since time

(3) **Handling NaN values**: NaN values were present only in Nominal attributes and were handled in two different ways as below.
1. Removal of rows having NaN values
2. Replacing NaN with the mode value of the attribute

(4) **Handling nominal attributes:**
One-hot encoding was performed for all the nominal attributes. All the values that occur less in than 100 instances (0.007 percent of the dataset) were given the same encoding.

(5) **Handling numeric attributes:**
L2 normalization was performed to all the numeric attributes.

(6) **Dimensionality reduction:**
Chi2 test was performed for all the attributes with the target attributes to identify the top 50 attributes having the highest chi2 score.

4 CLASSIFIERS IMPLEMENTED

(1) **Support Vector Machine**:
Support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary

linear classifier .

(2) **Random Forest Classifier**: Random forest consists of a large number of individual decision trees that operates as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

(3) **Linear Regression**: Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

(4) **Naive Bayes Classifier**: It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

(5) **Ensemble**: Ensemble learning seeks better predictive performance by combining the predictions from multiple models. We have combined Linear Regression, Random forest, Support vector machine and Naive Bayes Classifier to create a ensemble model for our project.

(6) **Cost sensitive weighted Random forest classifier**: This algorithm prevents performance drop due to imbalance in banking transaction dataset. While training, we divide dataset into overlapping samples. We train decision trees for each sample using C4.5 trainer and calculate error for each tree (equal importance to majority minority classes)and then break tie between trees using minority class error.

(7) **Modified Cost sensitive weighted Random forest classifier**: We modified the testing method of Cost sensitive weighted Random forest classifier by taking the weighted average of tree predictions.

(8) **XBNet classifier**: XBNet is the combination of tree based models and neural networks. While training, boosted Gradient Descent initialised with the feature importance of a gradient boosted tree. It updates the weights by gradient descent and then by using feature importance of the gradient boosted tree in every intermediate layer. Feature importance of an attribute is determined by the result of the difference between the information gain before the split and after the split this attribute in the tree While testing, an instance of the data is forward propagated through the trained model and the label for the instance is decided based on the output of the model.

5 RESULTS AND INSIGHTS:

The 4 datasets obtained by performing the initial data cleaning and preprocessing were tested against all the above mentioned algorithms

		Timestamp attributes retained		Timestamp attributes removed	
		NaN attributes removed	NaN replaced by mode	NaN attributes removed	NaN replaced by mode
Random forests	Training accuracy	91	89	90	88
	Testing accuracy	90	89	89	87
SVM	Training accuracy	80	64	75	58
	Testing accuracy	80	64	75	58
Naive Bayes	Training accuracy	86	84	84	82
	Testing accuracy	86	84	84	82
Logistic Regression	Training accuracy	86	82	83	82
	Testing accuracy	86	82	83	82
Ensemble	Training accuracy	87	86	85	84
	Testing accuracy	87	85	85	83
Cost-sensitive weighted random forest	Training accuracy	87	85	87	84
	Testing accuracy	88	86	87	86
Modified cost-sensitive weighted random forest	Training accuracy	92	89	90	88
	Testing accuracy	91	89	89	87
XBnet	Training accuracy	78	71	75	69
	Testing accuracy	82	76	70	77

Figure 3: Performance of different models

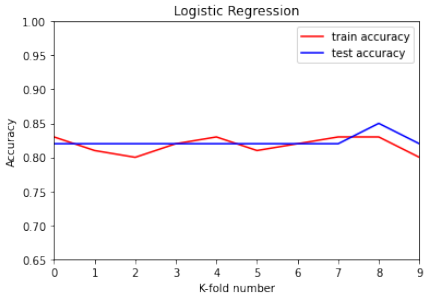

Figure 4: Accuracy for nan replaced by mode, timestamp removed, Logistic Regression

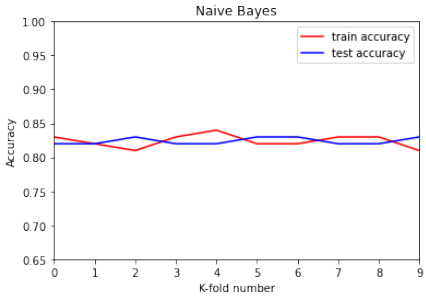

Figure 5: Accuracy for nan replaced by mode, timestamp removed, Naive Bayes Classification

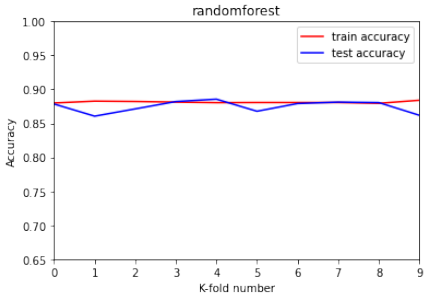

Figure 6: Accuracy for nan replaced by mode, timestamp removed, Random Forest

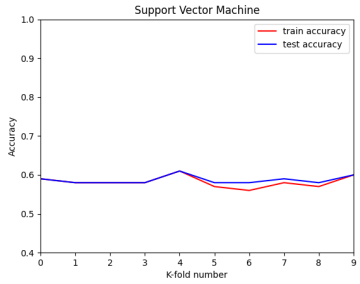

Figure 7: Accuracy for nan replaced by mode, timestamp removed, Support Vector Machine

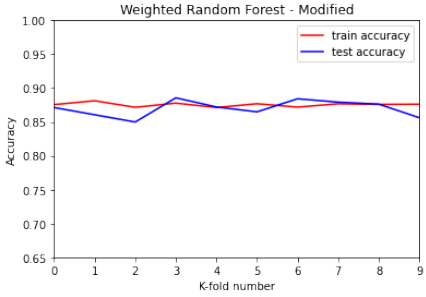

Figure 8: Accuracy for nan replaced by mode, timestamp removed, Modified cost Weighted RF

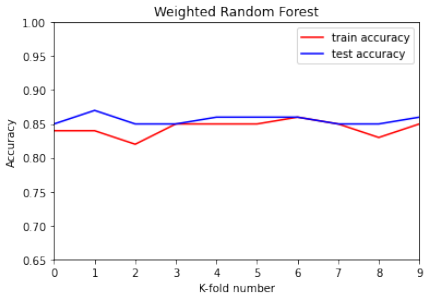

Figure 9: Accuracy for nan replaced by mode, timestamp removed, Cost weighted RF

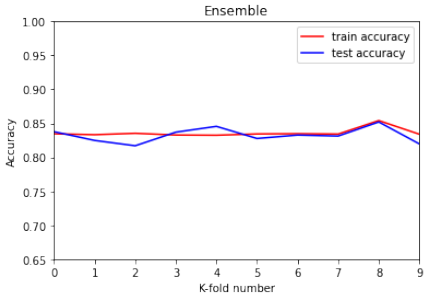

Figure 10: Accuracy for nan replaced by mode, timestamp removed, Ensemble Learning

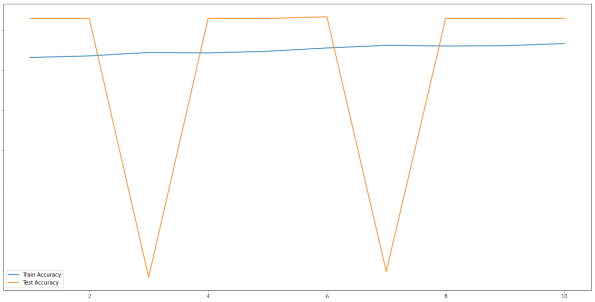

Figure 11: Accuracy for nan replaced by mode, timestamp removed, XBNet

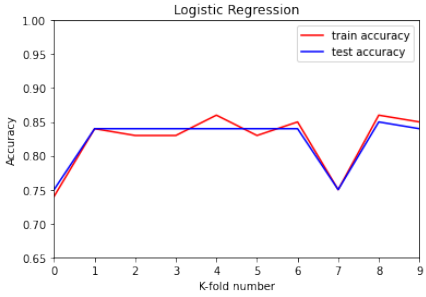

Figure 12: Accuracy for nan replaced by mode, timestamp retained, Logistic Regression

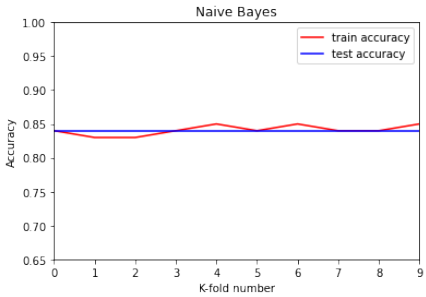

Figure 13: Accuracy for nan replaced by mode, timestamp retained, Naive Bayes Classification

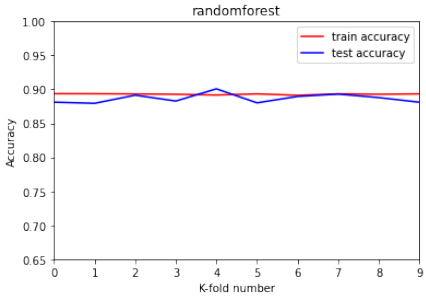

Figure 14: Accuracy for nan replaced by mode, timestamp retained, Random Forest

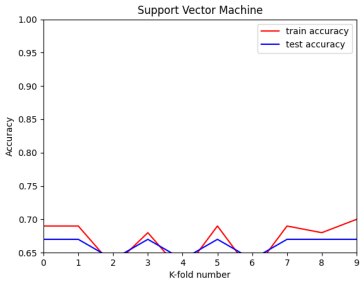

Figure 15: Accuracy for nan replaced by mode, timestamp retained, Support Vector Machine

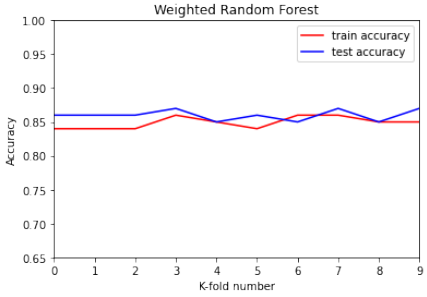

Figure 16: Accuracy for nan replaced by mode, timestamp retained, Cost weighted RF

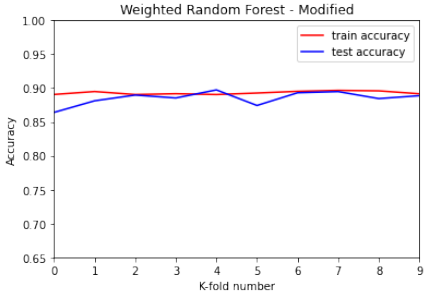

Figure 17: Accuracy for nan replaced by mode, timestamp retained, Modified cost weighted RF

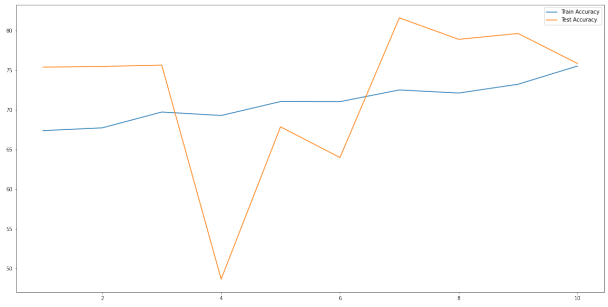

Figure 18: Accuracy for nan replaced by mode, timestamp retained, XBNet

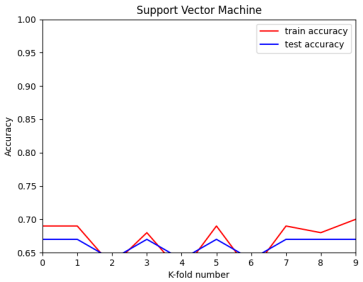

Figure 19: Accuracy for nan replaced by mode, timestamp retained, Support Vector Machine

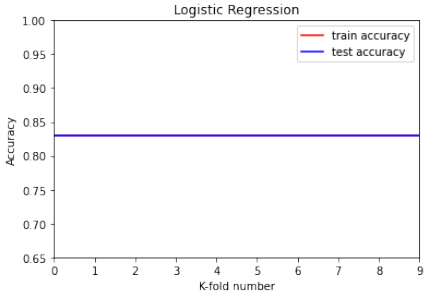

Figure 20: Accuracy for nan removed, timestamp removed, Logistic Regression

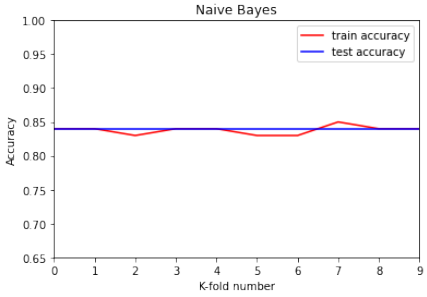

Figure 21: Accuracy for nan removed, timestamp removed, Naive Bayes Classification

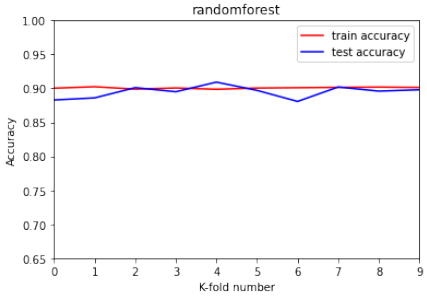

Figure 22: Accuracy for nan removed, timestamp removed, Random Forest

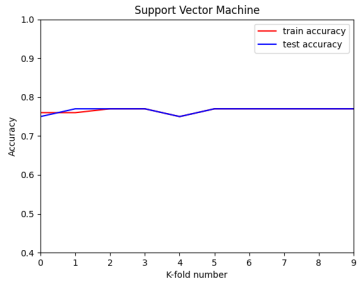

Figure 23: Accuracy for nan removed, timestamp removed, Support Vector Machine

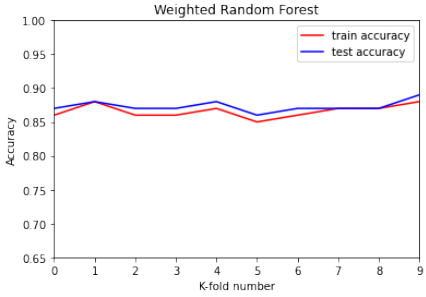

Figure 24: Accuracy for nan removed, timestamp removed, Cost Weighted RF

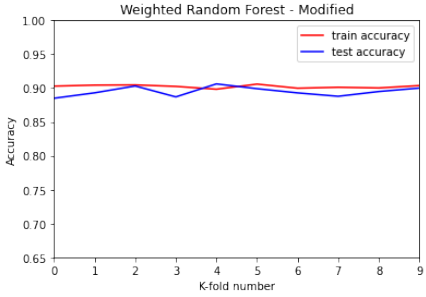

Figure 25: Accuracy for nan removed, timestamp removed, Modified Cost Weighted RF

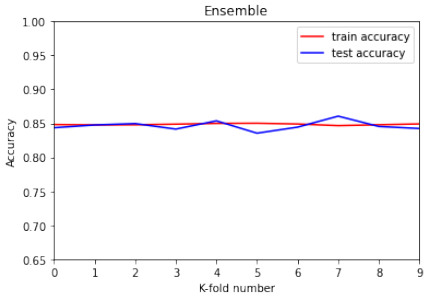

Figure 26: Accuracy for nan removed, timestamp removed, Ensemble Learning

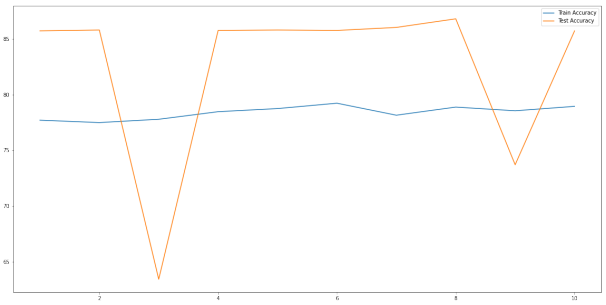

Figure 27: Accuracy for nan removed, timestamp removed, XBNet

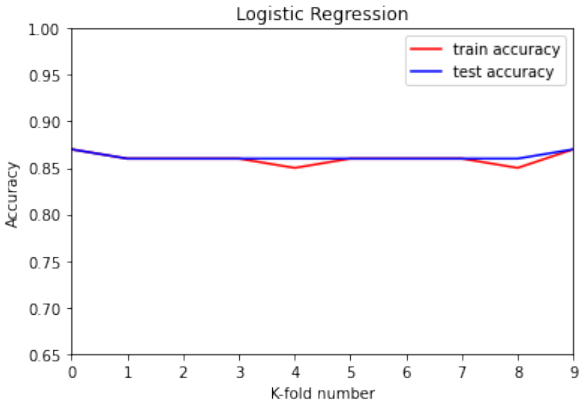

Figure 28: Removed nan,timestamp retained, Logistic Regression

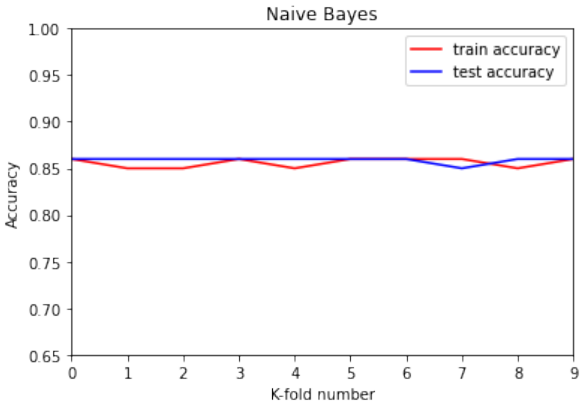

Figure 29: Removed nan,timestamp retained,Naive Bayes Classification

Github link: https://github.com/varshakvenkat/Elder-Fraud-Detection
Figures 3-18 shows the plots of training and testing accuracies of each model for each dataset on performing K-fold cross validation.
.

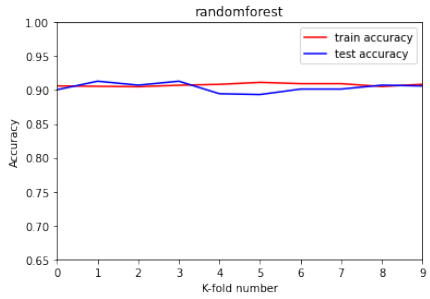

Figure 30: Removed nan,timestamp retained,Random Forest

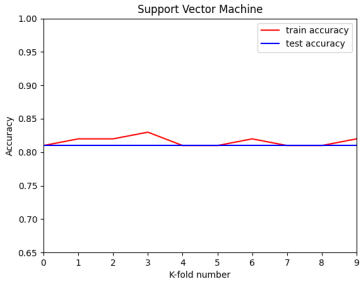

Figure 31: Removed nan,timestamp retained,Support Vector Machine

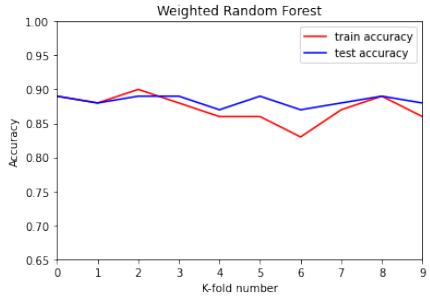

Figure 32: Removed nan,timestamp retained,Cost weighted RF

Figure 33: Removed nan,timestamp retained,Modified Cost weighted RF

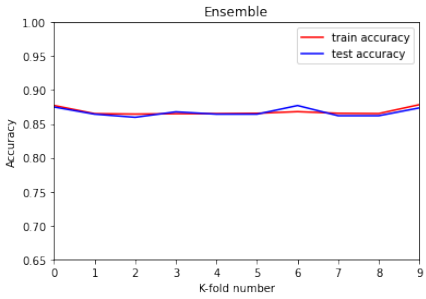

Figure 34: Removed nan,timestamp retained,Ensemble Learning

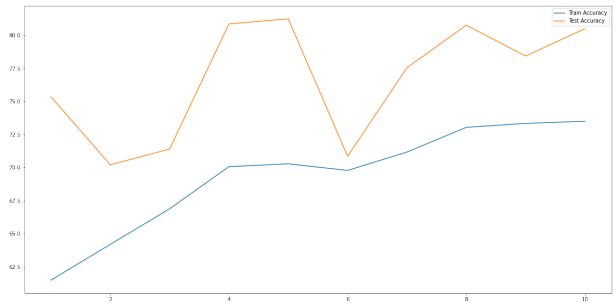

Figure 35: Removed nan, timestamp retained, XBNet

Removing NaN attributes gives a better accuracy than replacing them by the mode. Creation of new features based on the timestamps has led to an increase in the prediction accuracy. Modified cost sensitive weighted Random Forest gives highest training accuracy of 92% Testing accuracy of 91%. The modified cost weighted random forest method performed better than the original cost weighted random forest method that is mentioned in the literature survey because they are only considering only one tree completely ignoring the rest of tress while our modified version spreads the responsibility across multiple trees. Our model also performs better than XBNet. This could be due to the poor quality of the dataset. Additionally, XBNet makes use of deep neural network which requires a large amount of data to train. Our dataset is small and therefore XBNet model underfits.

6 EVALUATION OF THE OUTCOME OF YOUR PROJECT

We were able to achieve our goals for the final project as set in the proposal by implementing all the algorithms we proposed and testing them against the different versions of the dataset. Apart from the algorithms we had proposed to experiment on, we additionally implemented *Modified cost sensitive Random forest* which yielded us the best accuracy.

7 CONTRIBUTIONS

All members of the team contributed equally to the project.

REFERENCES

[1] Biswajit Purkayastha Debashree Devi, Saroj. K. Biswas. 2019. *A Cost-sensitive weighted Random Forest Technique for Credit Card Fraud Detection.*

[2] Lamiah Israt Fairoz Nower Khan, Amit Hasan Khan. 2020. *Credit Card Fraud Prediction and Classification using Deep Neural Network and Ensemble Learning.*

[3] Wells Fargo. 2021. *Campus Analytics Challenge 2021: Machine Learning Model to Predict Suspected Elder Fraud.* https://d18qs7yq39787j.cloudfront.net/uploads/contestfile/479/b765dc3d8076-trainset+%281%29.xlsx

[4] J. Gu. 2007. *Random Forest Based Imbalanced Data Cleaning and Classification.*

[5] Tushar Sarkar. 2021. *XBNet : An Extremely Boosted Neural Network.*