Machine Learning Pipeline for Fraud Detection and Prevention in E-Commerce Transactions

Resham Jhangiani
Department of Computer Science
California State University
Fullerton, California 92831
reshamjhangiani@csu.fullerton.edu

Doina Bein
Department of Computer Science
California State University
Fullerton, California 92831
dbein@fullerton.edu

Abhishek Verma
Department of Computer Science
New Jersey City University
Jersey City, NJ 07305
averma@njcu.edu

***Abstract*— Fraud has become a major problem in e-commerce and a lot of resources are being invested to recognize and prevent it. Present fraud detection and prevention systems are designed to prevent only a small fraction of fraudulent transactions processed, which still costs billions of dollars in loss. There is an urgent need for better fraud detection and prevention as the online transactions are estimated to increase substantially in the coming year. We propose a data driven model using machine learning algorithms on big data to predict the probability of a transaction being fraudulent or legitimate. The model was trained on historical e-commerce credit card transaction data to predict the probability of any future transaction by the customer being fraudulent. Supervised machine learning algorithms like Random Forest, Support Vector Machine, Gradient Boost and combinations of these are implemented and their performance are compared. While at the same time the problem of class imbalance is taken into consideration and techniques of oversampling and data pre-processing are performed before the model is trained on a classifier.**

Keywords— random forest; credit card fraud; support vector machine; gradient boost, logistic regression.

I. INTRODUCTION

Fraud can be defined as 'An act of intentional deception or dishonesty perpetrated by one or more individual, generally for financial gain'. With an ever-increasing use of the internet for shopping, banking, filing insurance claims etc., people and businesses have become targets of fraud in a whole new dimension. Reports claim that the growth in e-commerce fraud attempts in the first quarter of 2018 compared to 2016 outstripped the growth in e-commerce transactions by 83 percent. The E-commerce Fraud Index reported "Online department stores account takeover fraud increased from 0.06% in 2016 to 0.23% in 2017, representing more than 10% of total fraud losses". Increase in credit card fraud is limited to 0.1% of all card transactions, they have resulted in huge financial losses as fraudulent transactions have been large value transactions. However, with the increased use of credit card transactions in both volume and value over the recent years the proportion of fraud has remained the same.

We propose a data driven model using machine learning algorithms on big data to predict the probability of a transaction being fraudulent or legitimate. Our proposed model is trained on historical data. The model learns from data the signs, symptoms, trends and techniques used by fraudsters widely over the e-commerce platform. Based on its learning the model will be able to estimate the probability of a future or unseen transaction as fraudulent or legitimate. The model will also learn trends from the previous legitimate transactions, customer credit history to predict possibilities of customer performing any fraudulent transactions. Our proposed data driven model has accuracy while predicting a given transaction's probability of being fraudulent. Supervised machine learning algorithms like Random Forest, Support Vector Machine, Gradient Boost and combinations of these are implemented and their performance are compared. While at the same time the problem of class imbalance is taken into consideration and techniques of oversampling and data pre-processing are performed prior to training of the classification model.

The paper is organized as follows. In Section II we survey several representative techniques of machine learning and neural networks that are used for fraud detection and prediction. In Section III we describe the dataset used for benchmarking learned models. In Section IV we present our research methodology and proposed models. The performance of the models is given in Section V. Section VI presents experimental results. Concluding remarks and future work are given in Section VII.

II. RELATED WORK

At the beginning, fraud detection was identified with Information Retrieval or Rule-based approach. The information of each transaction was analyzed manually, and based on hard and fast rules transactions were flagged as fraudulent or legitimate. A feature vector is generated from every transaction that needs to be processed. A feature vector contains various parameters and attributes like TransactionID, TransactionAmount, Cardholderdetails, location of transaction, time of transaction. This feature vector is then scored with points, depending on pre-defined scoring rules set by human investigators. For example: "If previous transaction occurred in a different continent AND in less than one hour THEN fraud score=0.95" [1]. However this system relies on adding more and more rules to stay ahead of fraudsters that could exploit and circumnavigate existing rules.

The modern approach called Big Data Analytics via machine learning is more generalized, affordable, accurate and automated. A data-driven model is built that predicts, classifies, or estimates transaction being fraudulent or legitimate. This data driven model is trained on massive data generated from online datasets with large number of

transactions. Multiple approaches for Big Data Analytics were used to build various data-driven models. One of the following or combination of multiple approches have been used in existing work to build the model: Statistics and Information retrieval, Data mining, Machine learning (ML), Neural Networks (NN), and Fuzzy Logic.

The most widely used approach is to apply machine learning techniques to fit the data. Such data-driven model is more generalized and robust, which is the need of the current hour and provides accuracy as high as 87%. The most commonly used ML algorithms include: SVM, K-means, Regression Analysis, Decision Tree, Random Forest. Neural Network algorithms like Recurrent Neural Network (RNN), Long Short-Term Memory, Convolutional Neural Network sound promising for wide prediction classification tasks and can be applied to fraud detection and prevention as well.

Supervised machine learning approach is where the learning algorithm is first trained with data and labels, and later the accuracy is evaluated on test set. Supervised learning requires that data is labeled, before it is used for training the classifier; this process of labeling is highly expensive and time consuming. Classifiers like one class SVM [5], Decision Tree [6,7] Random Forest [1] and Logistic Regression [8] have proven to perform with a good accuracy. SVM can be used for both classification and regression [15]. One class SVM is of particular use where distribution of data is unbalanced; just like for our problem. It learns to infer the properties of the majority class and learn to detect anomalies or the minor class. Decision Trees are flow-chart like structures that lets you classify input data points or predict output given an input. A Random Forest is a robust approach to implement large number of decision trees and then ensemble their outputs. [2], [9].

Unlike supervised learning approach, unsupervised does not require labeled dataset. The algorithm makes inferences from the dataset without any labels. This technique has been widely adopted by the community because of the elimination of expensive data labeling process. The most adapted unsupervised algorithms for fraud analysis include Nearest neighbor, clustering and outlier detection.

Nearest neighbor uses a weighted measure to characterize and classify data points to be genuine or abnormal. Normal data instances cluster up together forming high density zones while outliers or anomaly data points are located far away from the dense zones. Local outlier factor has been used in [3]

Clustering uses similarities between data points to cluster them into groups. Similarities can be weighted difference of each data point from cluster centers. As for the fraud analysis data instances, which are legitimate, cluster up together based on various similarities in card holder profiles. Peer Group analysis [1] uses this approach.

Credit transactions are real-time streaming of millions of customer transactions throughout the globe. This data set is a timer based big dataset with high volume and velocity.

As the number of fraudulent transactions is much smaller than the legitimate once, the data distribution in unbalanced and this problem is termed as "class imbalance". It has been noticed that many learning algorithms underperform on unbalanced data; hence methods of resampling have to be adopted as prior to training the learning algorithm. In a fraud detection system, the massive number of payment request and transactions need to be labeled by human investigators before providing the dataset to the learning algorithm. The learning algorithm requires labeled datasets to first train itself and then update based on feedback provided by human investigators. This delay introduced into the system, due to restrain on number of transactions that can be evaluated by human investigators is termed as "verification latency".

Concept drift, which is tendency of transaction changing their statistical property overtime and sample selection bias, which is the difference between distribution of test and train sets are some other problems that need to be considered before modeling a machine learning algorithm for credit fraud analysis.

III. Dataset Description, Steps in Pre-processing

We propose a data-driven model to detect fraud in credit transaction. It needs to handle necessary data pre-processing and sampling techniques required for credit card imbalanced dataset. Next the model is trained with a machine learning algorithm to perform binary classification. In the process of finding the best machine learning algorithm various models are tested with multiple algorithms including SVM, Random Forest, Logistic Regression, Gradient Boost and Adaboost. The model must be flexible to accommodate big data, and predict with same or even better accuracy with the scale up of dataset. The multi-stage process for developing data-driven classification model is described in Fig. 1.

Historical data is extracted, processed, and cleaned. Features are explored and extracted. multiple machine learning algorithms are trained on this data. Each of these models is studied to identify cases of any overfitting or

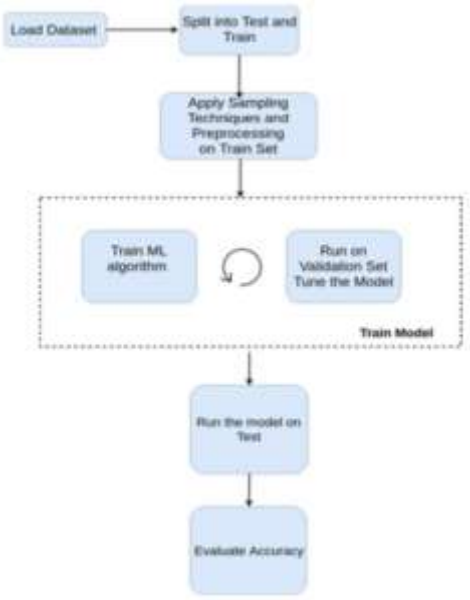

Fig. 1. The flow process in developing a machine learning model

underfitting. These models are further tuned or trained multiple times to remove underfitting or overfitting. These models are now tested on untrained or unseen data. The best fit model-the one with the highest accuracy is selected. The best fit model is then used for classification and over a

period of time, this model keeps learning and getting better in terms of accuracy.

The dataset used in the project is a public dataset donated by Machine Learning group of Université Libre de Bruxelles [10]. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numeric input variables, which are the result of a PCA transformation. Due to confidentiality issues, the original features and more background information about the data was not made available publicly. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. Card transaction dataset would also contain aggregated features. Some possible aggregated features include the average transaction amount per cardholder per month, average number of transactions per month, average expenditure on gas every month, time and location distance between last and current transaction, etc.

Fig. 2 depicts the dataset on a visual chart after applying the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm. t-SNE is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.

The dataset is split into test and train with a 70-30 distribution.

IV. Proposed Models

The number of fraud alerts raised by our proposed models would demand credit companies to review and process them. There is a restriction to the performance speed and resources that can be dedicated by such companies; that means system must only generate alarm when the confidence or the probability of transaction being fraud is high. At the same time it is preferred that a legitimate transaction is predicted false than a fraud predicted as legitimate. Over a period of time the credit card holder's nature of expenditure will change, accordingly there will be change in the transaction amount, number and frequency. Transaction will change their statistical property. Classification models should be able to adapt to such changes. It should have the possibility to scale up or down over the period of time as needed.

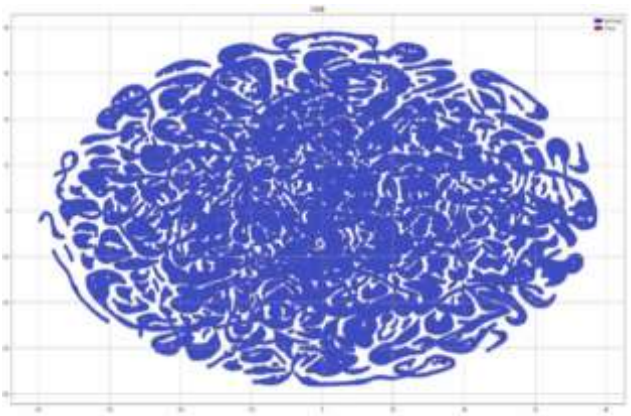

Fig. 2. Cluster view of the dataset with t-SNE visualization. Blue represents legitimate transactions. Tiny red dots represent fraudulent transactions.

Since the dataset is determined to be highly imbalanced, a machine learning algorithm running on imbalanced dataset would likely overfit for the majority class. Hence treating class imbalance is of high significance for building any good model.

The fraud detection system (FDS) must be capable enough to capture all these features that define the dataset. At the same time should be able to accommodate new features, or modify old features over the period of time. The current FDS model runs on 32 distinct features provided in the dataset.

To treat the problem of class imbalance, sampling techniques have been adopted. Sampling techniques are easy data level techniques that help resolve the imbalance problem. Sampling are balancing techniques used to balance the distribution between minority and majority class. Oversampling and undersampling are two types of sampling techniques. Those techniques either replicate or remove samples at random, there is no other information taken into consideration. Undersampling downsizes the majority class by randomly removing instances from majority class until dataset is balanced, whereas Oversampling randomly replicates instances of minority class to create balance. SMOTE is a special oversampling technique; it generates samples of minority class using interpolation.

We used both the oversampling and undersampling techniques to build the model. Model set 1 & 2 use random undersampling whereas model set 3 uses SMOTE oversampling technique. Details are shown in Table I.

TABLE I: Overview of the Proposed FDS Models

Model Set #	Sampling Technique	Data Pre-processing	ML Algorithm
1	Random Undersampling Dataset of size 1968	No	RF, Linear SVM, LR, Adaboost, XGboost, Pipeline
2	Random Undersampling Dataset size 984	Yes	RF, SVM, LR, Adaboost, XGboost, Pipeline
3	SMOTE Oversampling	Yes	RF, SVM, LR, Adaboost, XGboost, Pipeline

We used Random Forest (RF) from Scikit Learn Library; after tuning of n estimators, which is the number of decision trees in the forest, this value was set to 500. Random Forest seems to be a relatively stable algorithm for fraud detection analysis. Logistic Regression (LR) from Scikit Learn library was used to make the binary classification; the penalty parameter l2 regression was

chosen. Support Vector Machine (SVM) with the Linear Kernel Model from the Scikit Learn library was used to train the model, with the penalty parameter C set to default value of 1. Adaboost or Adaptive Boosting is an ensemble technique for Random Forest Classifier [1]. Gradient Boost is an ensemble of Random Forest and derived from Adaboost. The xgboost library is used for generating Gradient Boost algorithm. A Gradient Boost algorithm has three key components i.e. loss function, weak learner, and additive model. The additive model adds to the weak learner and the goal is to keep adding until loss is minimized. This has a much faster execution than RF. A pipeline of Logistic Regression model followed by Gradient Boost is applied. This is our proposed model for fraud analysis. We show later that this model always performs better with any of the sampling techniques applied.

We have used the following packages, tools and environments used for the project: Anaconda, Imbalanced Learn, Interactive Python, Jupyter Notebook, Matplotlib, Numpy, Pandas, Python 3.7, Scikit Learn, and Seaborn.

V. Performance of the Proposed Models

The performance of each model is measured and its accuracy evaluated. The performance measures adapted in this model are: Area under ROC, Precision Recall, and average precision:

Precision = True Positive / (True positive + False Positive)

Recall = True Positive / (True Positive + False Negative)

A precision-recall curve is a plot of the precision (y-axis) and the recall (x-axis) for different thresholds.

Average Precision (Avg. Prec.) summarizes the weighted increase in precision with each change in recall for the thresholds in the precision-recall curve. Smaller values on the x-axis of the plot indicate lower false positives and higher true negatives. Larger values on the y-axis of the plot indicate higher true positives and lower false negatives.

For the problem of class imbalance, the more relevant performance measure is not how well the classifier was able to predict majority class (0) correctly rather on how well the classifier predicts minority class (1) correctly. Thus we need to use precision-recall curve.

TABLE II. Results from six machine learning algorithms (Model Set 1)

ML Classifier	Avg. Prec.	AUC ROC	R-squared score
Random Forest	91.37	94.59	84.42
SVM	79.50	86.49	62.35
Logistic Regression	83.22	91.95	71.44
Gradient Boost	90.24	94.72	83.12
Pipeline (GBT+LR)	97.46	84.71	98.42
Adaboost with Random Forest	89.75	93.94	81.82

For FDS model set 1, undersampling technique is used. After the model is undersampled, it is trained on Random Forest, Linear SVM, Logistic Regression, Gradient Boost, proposed pipeline of Gradient Boost and Logistic Regression, and Adaboost. No other data preprocessing techniques is adopted. The performance measures opted are Precision, AUC ROC and R-squared. Table II gives a quick overview of the efficiency for all the different algorithms. Model set 2 uses undersampling with preprocessing, the results are in Table III.

TABLE III. Results from six machine learning algorithms for Model set 2

ML Classifier	Avg. Prec.	AUC ROC	R-squared score
Random Forest	70.15	90.31	68.50
SVM	22.38	85.77	-87.37
Logistic Regression	04.51	92.38	-15.16
Gradient Boost	35.45	92.5	-33.26
Pipeline (GBT+LR)	74.29	90.03	65.27
Adaboost with Random Forest	70.15	90.31	68.50

The FDS model set 3 uses the SMOTE oversampling technique. We plot the distribution of Transaction Amount and Transaction Time in the overall dataset (see Fig. 3). In Fig. 4 we plotted the Transaction Time independent of Amount. The first inference made was to eliminate the Time column from the feature vector. The second inference made is that since all other features V1 to V28 are scaled; Amount should also be scaled. We noticed that the amount of Fraud Transactions tends to be low, as shown in Fig. 5. Hence the decision of scaling.

TABLE IV. Results from six machine learning algorithms for Model set 3

ML Classifier	Avg. Prec.	AUC ROC	R-squared score
Random Forest	97.399	97.30	88.75
SVM	91.91	95.03	80.12
Logistic Regression	95.42	96.50	85.94
Gradient Boost	89.46	93.67	74.42
Pipeline (GBT+LR)	98.71	98.619	77.29
Adaboost with Random Forest	70.15	90.31	68.5

Some features in dataset may highly correlate with one another, so it is important to identify these correlations and, if possible, eliminate or normalize them. Based on inferences made during Model set 2, the Time column was dropped, and Amount column was scaled for the dataset used in Model set 3. These dataset samples were first shuffled and then randomly undersampled to get a dataset size of 492 instances. (As the original data set had only 492

Fig. 3 Distribution of Transaction Amount and Transaction Time in the overall dataset

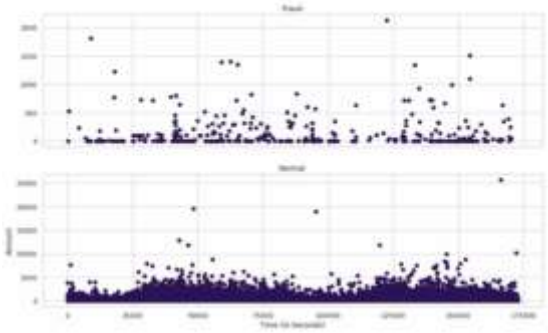

Fig. 4 Plot of the Transaction Time independent of Amount

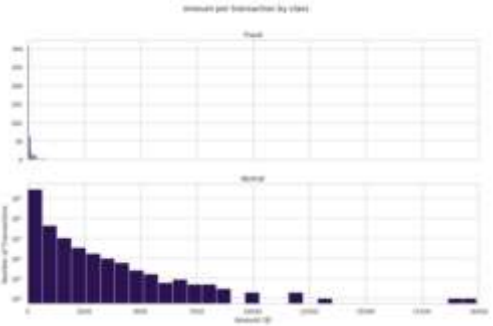

Fig. 5 Chart displaying the amount of Fraud Transactions

minority class samples.) Features V10, V12, V14, V17 have negative correlation with the class labels, whereas V2, V4, V11 V19 have positive correlation with the class label. For an FDS model, the negative correlation is more important. The outliers for V10, V12 and V14 will be removed, i.e those instances will be deleted from the dataset. V17 has not many outliers hence it is not modified. The dataset in now processed and undersampled as needed, thus the next step is to split into X and Y feature vectors and train the model. Table IV shows the efficiency of the machine learning algorithms on the current dataset.

VI. Discussion and Analysis of Research Results

The implementation of algorithms like RF, SVM, LR, GB, Pipeline and Adaboost on different processed and sampled datasets creates a huge difference in the accuracy, performance and time of execution. Sampling technique, outliers and correlation are some important factors to consider when developing a model. The performance of each FDS model cannot be compared with one another on different datasets or using different sampling technique and processing steps. Thus, we analyze our models among themselves and seek the ones that has the best classification performance in terms of high precision, high Kappa score and AUC ROC.

For FDS Model set 1, the performance measures for Random Forest and Gradient boost are very similar; the major difference in the two is the speed of execution. Gradient Boost was faster than RF; where RF completed the job in 10+/- 3 minutes, Gradient Boost achieved the same job under 4 minutes. Whereas the Linear SVM and Logistic Regression achieved low accuracy (Kappa score) of 79.4 and 85.45. SVM and LR can be avoided for fraud detection problem space. Undersampling did create an impact on the performance efficiency of the model. A simple Random Forest algorithm with n estimators set to 500 was run on the original dataset. Undersampling alone on this dataset increased the RF algorithm average precision from 73.13 to 91.36. The accuracy (Kappa score) increased from 84.8 to 91.97.

For FDS model set 2, SVM and LR are a bad fit since they had a very low precision and accuracy (Kappa score); these algorithms fail to learn and perform on the dataset. The fact that these two algorithms did not do good on undersampled data is worth for more research and analysis. The Random Forest and its derivative Adaboost performed almost the same and did okay in predicting the Fraud oversampled dataset. The accuracy (Kappa score) for the two was almost the same at 83.66, and precision at 70.15 for both. On the contrary the Gradient boost, which is derivative of both RF and Adaboost failed; with average precision as low as 35.44 and accuracy at 55.91.

The accuracy of all the algorithms compared to other models was highest for model set 3; however as mentioned before the three cannot be compared among themselves. RF did a pretty good job in evaluating the undersampled data with average precision of 97.39 and accuracy or Kappa score of 94.39. Clearly RF seems to work well for the fraud analysis dataset no matter what sampling and preprocessing techniques are implemented on dataset. RF also outperformed Gradient Boost, while the performance metrics for RF and Adaboost almost remained the same. It can be inferred that the ensemble/boosting techniques made no big difference in this model as the size of dataset 492 instances was very small. SVM and LR did a fair job on this dataset with average precision of 91.91 and 95.42.

We separately analyzed the proposed pipeline of Gradient Boost + Logistic Regression. The model works as follows: we first perform Gradient Boost algorithm on the dataset, the results of Gradient boost are then encoded using one hot encoder algorithm; and then fed into Logistic Regression algorithm for classification. This pipeline transfers the learning from gradient boost onto the logistic regression. The results of this pipeline during the implementation of three FDS are shown in Figures 6, 7, and 8.

VII. Conclusion and Future Work

Fraud Detection models presented in this paper solves the problem of imbalance in credit transaction dataset. Machine learning algorithms like Random Forest, Gradient Boost and Adaboost are good to operate on fraud transactions. Sampling techniques are easy and efficient in

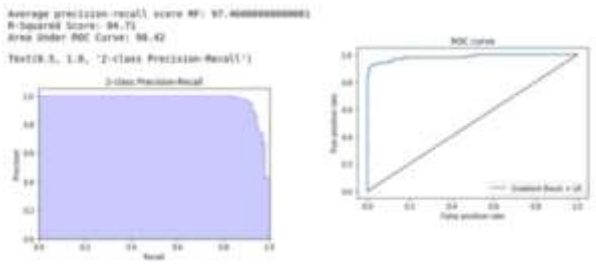

Fig. 6 Performance of Pipeline (GBT+LR) from model set 1.

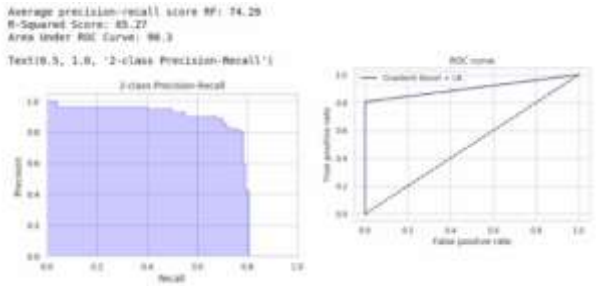

Fig. 7 Performance of Pipeline (GBT+LR) from model set 2.

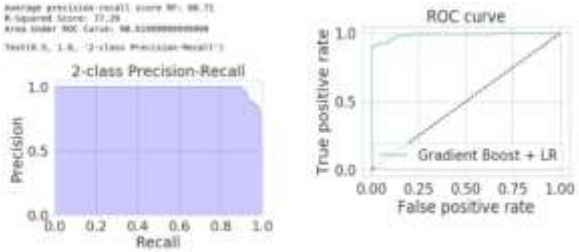

Fig. 8 Performance of Pipeline (GBT+LR) from model set 3.

resolving class imbalance problem. Machine learning algorithms like SVM and LR fail to perform well on such unbalanced binary classification problem. The proposed pipeline model looks promising with stable performance metrics.

The dataset used for this project implementation was limited as the data was transformed with PCA; as a result, a large information on nature of dataset; relation among the features were missing. Because of which very limited data preprocessing techniques could be implemented on the model. The scarcity of proper fraud transaction data is a major limitation on performing public research work and implementation in this domain. Further, the dataset used is recorded in the year 2013 for cardholders of Wordline; it is likely that since 2013 many changes in the nature, pattern of credit transactions and card holders along with the techniques to perform fraud may have evolved.

The models generated in this work would need few updates and tuning before implementing them in real world. ML classifiers Random Forest, AdaBoost and proposed pipeline of Gradient Boost + Logistic Regression will surely deliver high accuracy with right data preprocessing. Other important ML models worth considering in Fraud detection would be Active Learning and Light weight Neural nets with 3-4 hidden layers. With a more realistic dataset the FDS can be trained to resolve problems of concept drift and sample selection bias. Active learning is a recent addition to machine learning field; and the fact that it resolves problem of data labeling is good enough to explore active learning for fraud detection.

Lastly the software for FDS is a standalone application. Standalone applications are very limiting and hence a very important future extension on FDS would be to deploy this software as a service on cloud platform. REST API calls implementing the FDS functionalities and backend server running the machine learning algorithm will be deployed on cloud services. This will also eliminate the resource limitation of the model. To accomplish working of neural networks would require GPU computing, which could be easily managed on the cloud.

REFERENCES

[1] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi and G. Bontempi. Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy, IEEE Transactions on Neural Networks and Learning Systems (TNNLS), 2017.

[2] F. Carcillo, A. Dal Pozzolo, Y. Le Borgne, O. Caelen, Y. Mazzer and G. Bontempi. SCARFF: a Scalable Framework for Streaming Credit Card Fraud Detection with Spark, Information Fusion, Elsevier, 2017

[3] F. Carcillo, Y.A. Le Borgne, O. Caelen, G. Bontempi. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization, International Journal of Data Science and Analytics, Springer 2018

[4] A. Dal Pozzolo, O. Caelen, Y. Le Borgne, S. Waterschoot, and G. Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective, Expert Systems with Applications, vol. 41, no. 10, pp. 4915–4928, 2014.

[5] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams. Transaction aggregation as a strategy for credit card fraud detection, Data Mining and Knowledge Discovery, 18(1):30–55, 2009

[6] A. C. Bahnsen, D. Aouada, and B. Ottersten. Example-dependent cost sensitive decision trees, Expert Systems with Applications, 2015.

[7] A. Dal Pozzolo, R. A. Johnson, O. Caelen, S. Waterschoot, N. V. Chawla, and G. Bontempi. Using HDDT to avoid instances propagation in unbalanced and evolving data streams. In International Joint Conference on Neural Networks, pages 588–594. IEEE, 2014

[8] S. Jha, M. Guillen, and J. C. Westland. Employing transaction aggregation strategy to detect credit card fraud. Expert systems with applications, 39(16):12650–12657, 2012.

[9] Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, pp. 413–422 (2008)

[10] Machine Learning Group, University of Bruxelles, https://mlg.ulb.ac.be/wordpress/