Detection of Merchant Fraud using Machine Learning Techniques

Gadasu Sujith Kumar
sujith.mgit@gmail.com

Dr. Thayabba Khatoon
thayyaba.khatoon16@gmail.com

ABSTRACT

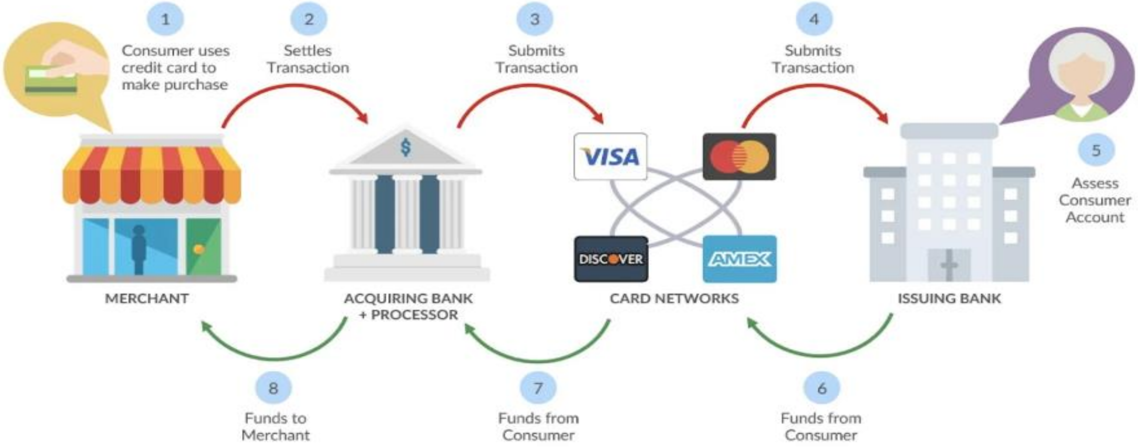
This project, Detection of Merchant Fraud using Machine Learning techniques is the process of monitoring customer behaviour and transactions to identify and prevent fraudulent activity. Transactions at Merchants are now potentially the most widely used in offline and online purchases. Due to new developments in electronic commerce systems and communication technology there is much more fraud involved with merchant transactions. Researchers face a difficult task when trying to identify Merchant services since criminals are quick-thinking and inventive. The increased use of electronic payments is now significantly impacted by the detection of fraudulent transactions. As a result, methods that are efficient and effective for identifying fraud in merchant services are required. Gradient Boosting Classifier, a machine learning methodology, is suggested in this research as a smart method for identifying fraud in merchant transactions. The performance of the proposed approach is evaluated based on real-world data sets.

1 INTRODUCTION

1.1 Introduction to Merchant Services

Merchant services is a broad term used to describe the range of financial services tailored to businesses. These services generally include financial tools like processing payments, payment gateways and even loyalty programs. Finding a solid merchant service provider that can help you take payments with ease is invaluable. These providers can support your business with intuitive software, efficient payment processing hardware and more that takes the hassle out of payment processing. Merchant services are a range of financial services and tools that businesses use to process payments. These services can include:

- Payment processing: Credit card, debit card, and other electronic payment methods
- Check processing: Check guarantee and conversion services, and automated clearing house check drafting and payment services
- Gift cards: Gift cards and loyalty programs
- Payment gateways: Authorize the processing of card or direct payments
- Technology integration: Transaction tracking technology integration
- POS services: Credit card terminals and mobile card readers



Nowadays, most organizations, companies and government agencies have adopted electronic commerce to increase their productivity or efficiency in trading products or services; in areas such as credit card, telecommunication, healthcare insurance, automobile insurance, online auction, etc. Electronic commerce systems are used by both legitimate users and fraudsters; hence they become more vulnerable to large scale and systematic fraud.

1.2 What is Fraud ?

Fraud refers to deceptive or illegal activities that individuals or groups engage into exploiting weakness in a payment processing system for financial gain. Fraud is a crime where the purpose is to appropriate money by illegal means. The Association of Certified Fraud Examiners (ACFE) defines “fraud” as: the use of one's occupation for personal enrichment through the deliberate misuse or misapplication of the employing organization's resources or assets. Internet Crime Complaint Centre (IC3) is a valuable resource for both victims of Internet crime and law enforcement agencies in identifying, investigating and prosecuting these crimes.

In 2014, the IC3 received 269,422 complaints with an adjusted dollar loss of $800,492,073; which is a 2.39 percent increase in reported losses since 2013 ($781,841,611) (IC3, 2014). The number of complaints received by the IC3 between 2011 and 2014 and the corresponding dollar losses. From this table, amount of loss steadily increase while number of complaints decrease; this is because, fraud is causing more loss now compared to the past. These huge number of losses have increased the importance of fraud fighting. The purpose of fraud prevention mechanism is to protect the technological systems against fraud by stopping fraud from occurring in the first place. Nevertheless, this mechanism alone is not enough to halt fraud. Fraud detection is also proposed to improve the technological systems security.

1.3 Fraud Detection

Fraud detection is the process of monitoring customer behavior and transactions to identify and prevent fraudulent activity. Fraud detection detects and recognizes fraudulent activities as they enter the systems and reports them to a system administrator. Similar to detection approaches in Intrusion detection system (IDS), Fraud detection system also uses misuse and anomaly based approaches to detect fraud. Both misuse based FDS and anomaly based FDS utilize data mining techniques to determine fraud from large amount of incoming data stream. However, there are issues and challenges that hinder the development of an ideal FDS for E-commerce system; such as concept drift, supports real time detection, earliness of detection, skewed distribution, large amount of data, Incorrect classification cost, etc. The presence of any one of these challenges will lead to high false alerts, low detection accuracy and slow detection. These are the parameters used to characterize the performance of FDS. In this paper, we will survey fraud detection systems on merchant services.

People can use online transactions at Merchant as it provides an efficient and easy-to-use facility. With the increase in usage of merchant services, the capacity of misuse has also enhanced. Merchant services frauds cause significant financial losses for both card holders and financial companies. In this research study, the main aim is to detect such frauds, including the accessibility of public data, high-class imbalance data, the changes in fraud nature, and high rates of false alarm.

1.4 Fraud detection issues and challenges

Fraud detection is a complex domain; we may find that a fraud detection system is prone to fail, has a low accuracy rate, or gives many false alarms. It is extremely difficult for electronic commerce systems to handle fraud problem forcing them to incur heavy losses. This happens because fraud detection systems need to deal with multiple challenges to be taken into account. Several challenging properties that fraud detection must deal with will be presented in this section. Fig. 3 shows

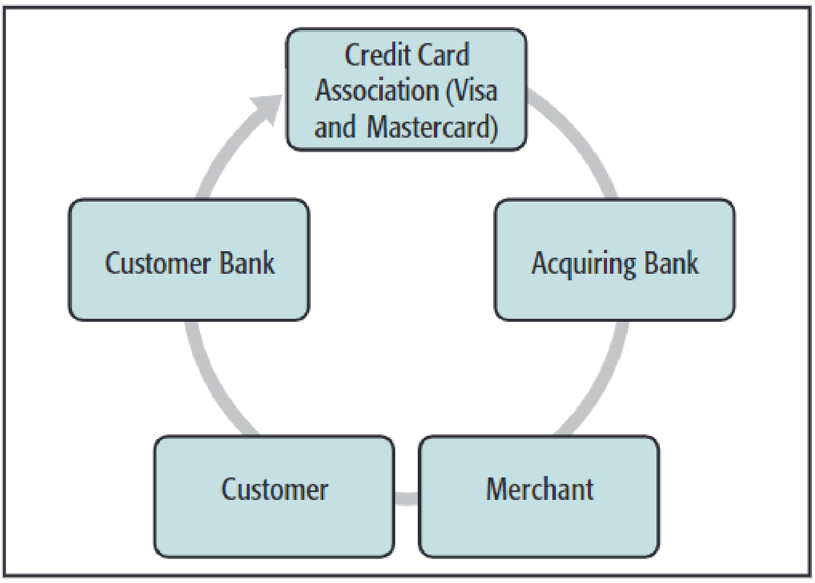
1.5 Fraud areas

Almost any technological system that involves money and services can be compromised by fraudulent acts, for example credit card system, telecommunication system, health care insurance system, etc. (Almeida et al., 2008). This section is going to address fraud happening to the five most prevalent areas, which are credit card, telecommunication, health care insurance, automobile insurance and online auction areas.

1.6 Merchant Acquirer Relationship

Business Framework

Merchants, who accept credit cards, get a Point of Sale(POS) device installed at their outlet. This POS device is installed through a bank. And this bank is known as the acquiring bank, or simply the acquirer. When a transaction is recorded by a merchant, a copy of the receipt signed by the customer is sent to the bank by the merchant for claiming the amount of transaction. The acquiring bank settles this within a stipulated number of days, after deducting a surcharge, known as discount rate. Figure 1 shows the flow of events once a transaction is done by a customer at a merchant. The merchant sends a copy of the signed receipt to the acquiring bank. The bank,in turn, settles the dues to the merchant, and also communicates the transaction details to the customer bank, via the Credit Card Association. Once this communication is received by the customer bank, it adds this transaction to the customer’s statement of account.



Approach to Problem statement

The relevant literature presents many machines learning based approaches for merchant fraud detection, such as Extreme Learning Method, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression and XG Boost. However, due to low accuracy, there is still a need to apply state of the art deep learning algorithms to reduce fraud losses. The main focus has been to apply the recent development of deep learning algorithms for this purpose. Comparative analysis of both machine learning and deep learning algorithms was performed to find efficient outcomes. The detailed empirical analysis is carried out using the dataset for fraud detection. A machine learning algorithm was first applied to the dataset, which improved the accuracy of detection of the frauds to some extent. Later, three architectures based on a convolutional neural network are applied to improve fraud detection performance. Further addition of layers further increased the accuracy of detection. The proposed model outperforms the state-of-the-art machine learning and deep learning algorithms for merchant fraud detection problems. In addition, we have performed experiments by balancing the data and applying deep learning algorithms to minimize the false negative rate. The proposed approaches can be implemented effectively for the real-world detection of merchant fraud.

2. LITERATURE REVIEW

In Literature Review, different techniques have been proposed for Fraud detection. This section reviews the existing techniques for Fraud prevention and detection. Some research issues arising out of the review of existing methods are identified, and addressed.

2.1 Fraud Detection based on Auto Encoders

Credit Card Fraud detection based on Deep Learning (Y. Abakarim, M. Lahby, and A. Attioui)

In the last decades Machine Learning achieved notable results in various areas of data processing and classification, which made the creation of real-time interactive and intelligent systems possible. The accuracy and precision of those systems depends not only on the correctness of the data, logically and chronologically, but also on the time the feed-backs are produced. This paper focuses on one of these systems which is a fraud detection system. In order to have a more accurate and precise fraud detection system, banks and financial institutions are investing more and more today in perfecting the algorithms and data analysis technologies used to identify and combat fraud. Therefore, many solutions and algorithms using machine learning have been proposed in literature to deal with this issue. However, comparison studies exploring Deep learning paradigms are scarce, and to our knowledge, the proposed works don't consider the importance of a Real-time approach for this type of problems. Thus, to cope with this problem we propose a live credit card fraud detection system based on a deep neural network technology. Our proposed model is based on an auto-encoder and it permits to classify, in real-time, credit card transactions as legitimate or fraudulent. To test the effectiveness of our model, four different binary classification models are used as a comparison. The Benchmark shows promising results for our proposed model than existing solutions in terms of accuracy, recall and precision.

2.2 Risk Assessment using RUSBoost

User authorization from imbalanced data logs of credit cards using artificial intelligence(V. Arora, R. S. Leekha, K. Lee, and A. Kataria)
An effective machine learning implementation means that artificial intelligence has tremendous potential to help and automate financial threat assessment for commercial firms and credit agencies. The scope of this study is to build a predictive framework to help the credit bureau by modelling/assessing the credit card delinquency risk. Machine learning enables risk assessment by predicting deception in large imbalanced data by classifying the transaction as normal or fraudster. In case of fraud transaction, an alert can be sent to the related financial organization that can suspend the release of payment for particular transaction. Of all the machine learning models such as RUSBoost, decision tree, logistic regression, multi-layer perceptron, K-nearest neighbor, random forest, and support vector machine, the overall predictive performance of customized RUSBoost is the most impressive. The evaluation metrics used in the experimentation are sensitivity, specificity, precision, F scores, and area under receiver operating characteristic and precision recall curves. Datasets used for training and testing of the models have been taken from kaggle.com.

2.3 Performance Analysis using Four Filter Ranking

Performance analysis of feature selection methods in software defect prediction: A search method approach (A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim)
Software Defect Prediction (SDP) models are built using software metrics derived from software systems. The quality of SDP models depends largely on the quality of software metrics (dataset) used to build the SDP models. High dimensionality is one of the data quality problems that affect the performance of SDP models. Feature selection (FS) is a proven method for addressing the dimensionality problem. However, the choice of FS method for SDP is still a problem, as most of the empirical studies on FS methods for SDP produce contradictory and inconsistent quality outcomes. Those FS methods behave differently due to different underlining computational characteristics. This could be due to the choices of search methods used in FS because the impact of FS depends on the choice of search method. It is hence imperative to comparatively analyze the FS methods performance based on different search methods in SDP. In this paper, four filter feature ranking (FFR) and fourteen filter feature subset selection (FSS) methods were evaluated using four different classifiers over five software defect datasets obtained from the National Aeronautics and Space Administration (NASA) repository. The experimental analysis showed that the application of FS improves the predictive performance of classifiers and the performance of FS methods can vary across datasets and classifiers. In the FFR methods, Information Gain demonstrated the greatest improvements in the performance of the prediction models. In FSS methods, Consistency Feature Subset Selection based on Best First Search had the best influence on the prediction models. However, prediction models

based on FFR proved to be more stable than those based on FSS methods. Hence, we conclude that FS methods improve the performance of SDP models, and that there is no single best FS method, as their performance varied according to datasets and the choice of the prediction model. However, we recommend the use of FFR methods as the prediction models based on FFR are more stable in terms of predictive performance.

2.4 Case study of Corruption control

Fraud and corruption control at education system level: A case study of the Victorian department of education and early childhood development in Australia (B. Bandaranayake)
This case describes the implementation of a fraud and corruption control policy initiative within the Victorian Department of Education and Early Childhood Development (the Department) in Australia. The policy initiative was administered and carried out by a small team of fraud control officials, including the author of this article, in the Department. The policy context represents a large, devolved and fragmented governance and accountability system. This case highlights the complexity of the policy initiative, the contextual constraints that challenged the implementation, and the pragmatic approach taken by the Department. While there are no easy solutions for fraud and corruption control or proven models to follow, this case presents helpful lessons for the professionals working in large and devolved education systems.

2.5 Fraud detection using Neural Networks

Credit card fraud detection model based on LSTM recurrent neural networks (I. Benchaji, S. Douzi, and B. E. Ouahidi)
With the increasing use of credit cards in electronic payments, financial institutions and service providers are vulnerable to fraud, costing huge losses every year. The design and the implementation of efficient fraud detection system is essential to reduce such losses. However, machine learning techniques used to detect automatically card fraud do not consider fraud sequences or behavior changes which may lead to false alarms. In this paper, we develop a credit card fraud detection system that employs Long Short-Term Memory (LSTM) networks as a sequence learner to include transaction sequences. The proposed approach aims to capture the historic purchase behavior of credit card holders with the goal of improving fraud detection accuracy on new incoming transactions. Experiments show that our proposed model gives strong results and its accuracy is quite high.

2.6 Top Algorithms for Fraud Detection

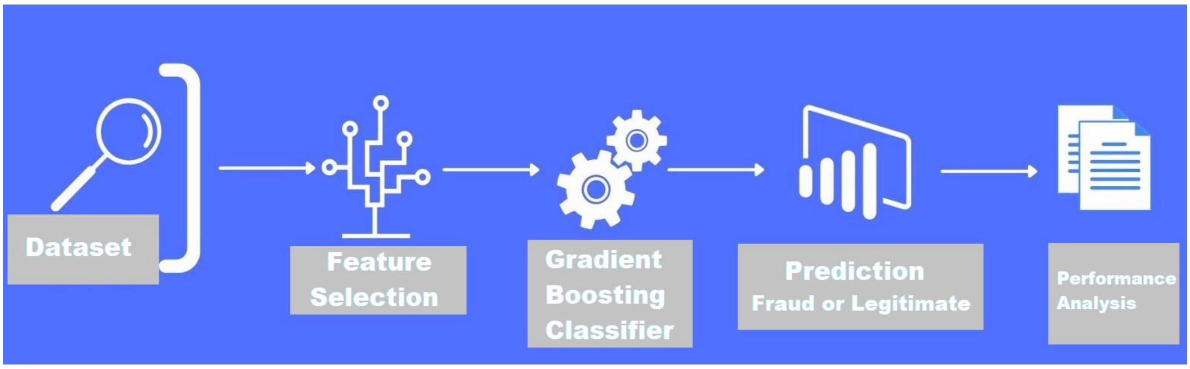
TOP 10 ALGORITHMS IN MACHINE LEARNING FOR FRAUD DETECTION
In the study, the top ten ML algorithms are incorporated for the detection of credit card frauds. The list of these algorithms is given below:
1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. SVM
5. Naïve Bayes
6. CNN
7. K-Means
8. Random Forest

9. Dimensionality Reduction Algorithms
10. Gradient Boosting Algorithms

These algorithms can also encompass association analysis, clustering, classification, statistical learning, and link mining.

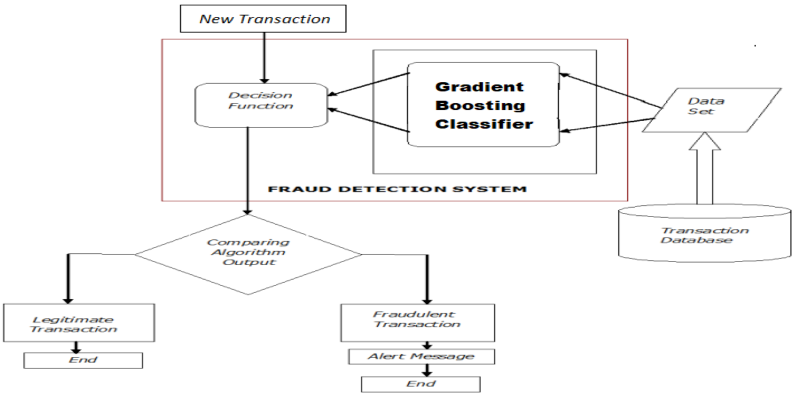
2.7 System Architecture

The proposed system which has Gradient boosting classifier can be more accurate than the existing system which uses such as random forests. Because we train them to correct each other's errors, they're capable of capturing complex patterns in the data.



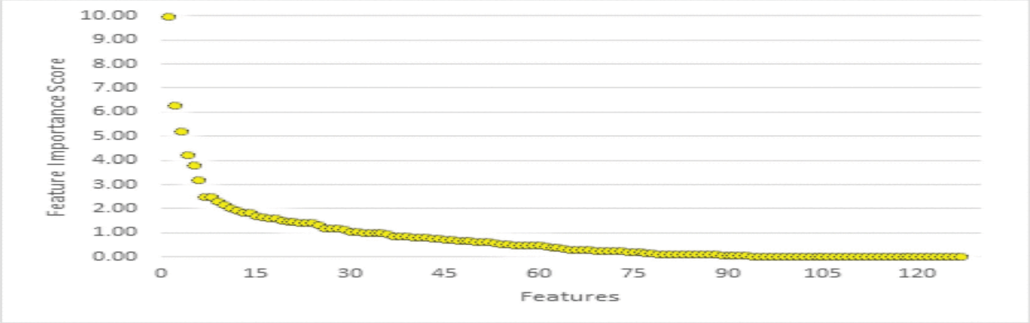
3 Problem Statement

There are multiple key factors that needed to be considered when designing the fraud detection system architecture. Detection frequency determines how often we run the new data through our fraud scoring model. Fraud-prevention operation flow impacts how and when we flag different events as suspicious, and how to handle and confirm those suspicious cases afterwards. Scoring accuracy baseline helps us to assess the qualification of our fraud scoring model.



3.1 Feature Selection

Generally, when we have a big model with hundreds or thousands of features, the feature selection approach is used to choose the most promising features and to remove irrelevant features while retraining the model. Also, by analyzing the importance of each feature manually, we can get an idea of what the model is doing, and the model is working well. Here, we derive the importance of each feature by applying WFI scoring method on Gradient Boosting trained model. Furthermore, all the features are depicted as a percentage rating of how often the feature is used in determining the output label. To make the list of features easier to read, we have sorted them from most important to least important as shown in Figure



The feature importance scores reflect information gain by each feature during the construction of a decision tree. During experiments, we observe 50% of the 128 features are not contributing to making any decision. The WFI score of such features is zero. While, out of the remaining 50% of features, 15 features provide a significant contribution in making decisions during the construction of decision-tree. The WFI score of those features has high values in the range of 1 to 10. The rest of the 45 features having feature importance scores between 0 and 1. These 45 additional features contribute comparatively less and have a large drop in feature importance score. Altogether the entire dataset is divided into three levels of information gain groupings, namely, most promising, slightly contributing, and irrelevant features.

Feature extraction creates a subset of the given features which not only reduces the noise but also improves the classifiers' performance. Therefore, we have tested 15 datasets of four different categories (binary, three-class, seven-class & Multi-class) of power grid system created by the Oak Ridge National Laboratories using the most promising features. To identify these best features, we use the WFI scoring model along with concept of Num trees.

Furthermore, to increase the execution speed, we perform feature extraction on binary datasets. We repeat the entire process by taking the various parameter value of Num_trees to collect various observations. From that we have identified best features by taking common important features from the estimations. Here, Num_trees refers to the number of estimators whereas n refers to the total number of features. We have used four estimators, namely, 100, 500, 700, 1000 and initially dataset consist of n= 128 features.

3.2 Gradient Boosting Classifier

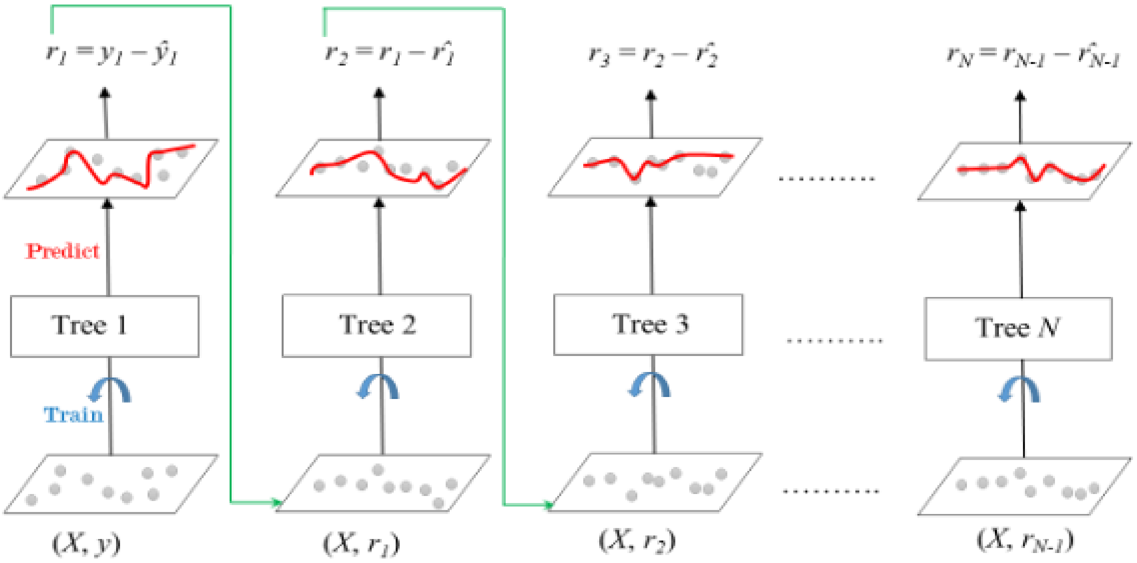
Gradient Boosting is an ensemble technique. It is primarily used in classification and regression tasks. It provides a forecast model consisting of a collection of weaker prediction models, mostly decision trees. Gradient Boosting is a type of machine learning boosting technique. It builds a better model by merging earlier models until the best model reduces the total prediction error. Also referred to as a statistical forecasting model, the main idea of gradient boosting is to attain a model that eliminates the errors of the previous models.

Gradient Boosting is named so that the set target outcomes depend on the gradient of the inaccuracy vs the forecast. Every new model created using this method moves closer to the path that lowers prediction error in the range of potential outcomes for every ML training case.

Gradient Boosting is mainly of two types depending on the target columns:

- Gradient Boosting Regressor: It is used when the columns are continuous
- Gradient Boosting Classifier: It is used when the target columns are classification problems

Let us understand how Gradient Boosting algorithms work through an example. In this case, we will have a continuous target column, thus using Gradient Boosting Regressor .
We will use a random dataset with different features. We have to predict target values, while all other characteristics are standalone features.
We need to observe if the learning algorithm is able to figure out the irrelevant characteristics.



4 IMPLEMENTATION

4.1 MODULES

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Analyze and Prediction
- Accuracy on test set
- Saving the Trained Model

4.2 MODULES DESCSRIPTION:

Data Collection:

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.
There are several techniques to collect the data, like web scraping, manual interventions and etc. Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms
Kaggle Link: https://www.kaggle.com/datasets/

Dataset:

The dataset consists of 1000 individual data. There are 21 columns in the dataset, which are described below.
Over_draft : An overdraft account allows users to withdraw more than they have in their bank account.
credit_usage : Users credit usage
credit_history : Users credit history
purpose : Users purpose
current_balance : Users current balance
Average_Credit_Balance : Users average credit balance
employment : Employment types
location : Users location
personal_status : Users personal status
other_parties : Other parties
residence_since : Users Residence
property_magnitude: Users Property
cc_age : Users age
other_payment_plans : payment plans
housing : housing types
existing_credits : users Existing credits
job : job types
num_dependents : Numbers of dependents
own_telephone : Users telephone
foreign_worker : Foreign worker yes or no
Class: Good or Bad

Data Preparation:

Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.)
Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data
Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis
Split into training and evaluation sets

Model Selection:

We used Gradient Boosting Classifier machine learning algorithm, We got a accuracy of 91 % on test set so we implemented this algorithm.

Gradient Boosting Classifier Algorithm:

In Gradient Boosting, each predictor tries to improve on its predecessor by reducing the errors. But the fascinating idea behind Gradient Boosting is that instead of fitting a predictor on the data at each iteration, it actually fits a new predictor to the residual errors made by the previous predictor. Let’s go through a step by step example of how Gradient Boosting Classification Works:
In order to make initial predictions on the data, the algorithm will get the log of the odds of the target feature. This is usually the number of True values (values equal to 1) divided by the number of False values (values equal to 0).

For every instance in the training set, it calculates the residuals for that instance, or, in other words, the observed value minus the predicted value.

Once it has done this, it builds a new Decision Tree that actually tries to predict the residuals that was previously calculated. However, this is where it gets slightly tricky in comparison with Gradient Boosting Regression.

When building a Decision Tree, there are a set number of leaves allowed. This can be set as a parameter by a user, and it is usually between 8 and 32. This leads to two of the possible outcomes:

- Multiple instances fall into the same leaf
- A single instance has its own leaf

Unlike Gradient Boosting for Regression, where we could simply average the instance values to get an output value, and leave the single instance as a leaf of its own, we have to transform these values using a formula:

$$\frac{\sum Residual}{\sum [PreviousProb * (1 - PreviousProb)]}$$

Credit: Blogspace

The Σ sign means "sum of", and PreviousProb refers to our previously calculated probability (in our example, being 0.7). We apply this transformation for every leaf in the tree. Why do we do this? Because remember our base estimator is a log (odds), and our tree was actually built on a probability, so we cannot simply add them because they come from two different sources.

Making Predictions

Now, to make new predictions, we do 2 things: get the log(odds) prediction for each instance in the training set convert that prediction into a probability

For each instance in the training set, the formula for making predictions would be the following:

base_log_odds + (learning_rate * predicted residual value)

The learning_rate is a hyper parameter that is used to scale each trees contribution, sacrificing bias for better variance. In other words, we multiply this number by the predicted value so that we do not overfit the data.

Once we have calculated the log(odds) prediction, we now must convert it into a probability using the previous formula for converting log(odds) values into probabilities

A gradient boosting classifier is used when the target column is binary. All the steps explained in the Gradient boosting regressor are used here, the only difference is we change the loss function. Earlier we used Mean squared error when the target column was continuous but this time, we will use log-likelihood as our loss function.

Let's see how this loss function works, to read more about log-likelihood I recommend you to go through this where I have given each detail you need to understand this.

The loss function for the classification problem is given below:

$$L = -\sum_{i=1}^{n} y_i \log(p) + (1-p)\log(1-p)$$

Our first step in the gradient boosting algorithm was to initialize the model with some constant value, there we used the average of the target column but here we'll use log(odds) to get that constant value. The question comes why log(odds)?

When we differentiate this loss function, we will get a function of log(odds) and then we need to find a value of

log(odds) for which the loss function is minimum.
let’s see how it works:
Let’s first transform this loss function so that it is a function of log(odds), I’ll tell you later why we did this transformation.

$$\text{Hence ,} \quad -\mathrm{y} * log(\mathrm{odds}) - \left(-log\left(1 + \mathrm{e}^{\log(\mathrm{odds})}\right)\right)$$

$$\mathrm{L} \;=\; -\mathrm{y} * log(\mathrm{odds}) + log\left(1 + \mathrm{e}^{\log(\mathrm{odds})}\right)$$

Now this is our loss function, and we need to minimize it, for this, we take the derivative of this w.r.t to log(odds) and then put it equal to 0,.

Analyze and Prediction:
In the actual dataset, we chose only 14 features :

credit_usage : Users credit usage
credit_history : Users credit history
purpose : Users purpose
current_balance : Users current balance
Average_Credit_Balance : Users average credit balance

personal_status : Users personal status
other_parties : Other parties
property_magnitude: Users Property
cc_age : Users age
other_payment_plans : payment plans
housing : housing types
job : job types
num_dependents : Numbers of dependents
foreign_worker : Foreign worker yes or no
Class: Good or Bad

Accuracy on test set:
We got an accuracy of 91.7% on test set.

Saving the Trained Model:

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle.
Make sure you have pickle installed in your environment.

Next, let’s import the module and dump the model into .pkl file.

CONCLUSION

The prevention of Merchant fraud is essential for increased transactions. The financial losses suffered by financial institutions are significant and ongoing, and the detection of merchant fraud is becoming more challenging, thus it is crucial to create more efficient methods for doing so. Gradient Boosting Classifier is used in this paper to suggest an intelligent method for identifying fraud in merchant transactions. We performed a number of experiments utilizing actual data. Performance analysis metrics were used to assess the performance of the suggested approach. According to the experimental findings, the suggested method outperformed other machine learning algorithms and attained the maximum accuracy performance. The outcomes demonstrate that the suggested method outperforms alternative classifiers. The outcomes further emphasize the significance and benefit of implementing an effective parameter optimization strategy for boosting the suggested approach's predictive capabilities.

References

[1] Y. Abakarim, M. Lahby, and A. Attioui, ``An effficient real time model for credit card fraud detection based on deep learning,'' in Proc. 12th Int. Conf. Intell. Systems: Theories Appl., Oct. 2018, pp. 17, doi: 10.1145/3289402.3289530.

[2] H. Abdi and L. J. Williams, ``Principal component analysis,'' Wiley Interdiscipl. Rev., Comput. Statist., vol. 2, no. 4, pp. 433459, Jul. 2010, doi: 10.1002/wics.101.

[3] V. Arora, R. S. Leekha, K. Lee, and A. Kataria, ``Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence,'' Mobile Inf. Syst., vol. 2020, pp. 113, Oct. 2020, doi: 10.1155/2020/8885269.

[4] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, ``Performance analysis of feature selection methods in software defect prediction: A search method approach,'' Appl. Sci., vol. 9, no. 13, p. 2764, Jul. 2019, doi: 10.3390/app9132764.

[5] B. Bandaranayake, ``Fraud and corruption control at education system level: A case study of the Victorian department of education and early childhood development in Australia,'' J. Cases Educ. Leadership, vol. 17, no. 4, pp. 3453, Dec. 2014, doi: 10.1177/1555458914549669.