Smart Credit Card Fraud Detection Approach Using Supervised Machine Learning Techniques

Pankaj S.Patel[1], Santosh Maheshwari[2], Vatsala D.Patel[3], Dharmesh Shah[4]
[1, 2]Hansaba College of engineering Sidhdhpur
[3, 4] SSPC Visnagar

Abstract- *According to the trends the use of credit card for shopping and payment by people is increasing effectively. Receiving the details of the card and exchanging money is called extortion or we can say shakedown we can see more and more frequently. Safety measures and precautions are mainly desirable due to basic monitory in the various types of industries. In this research showing the detecting a fraud in credit card transaction with the use of Naïve Bayes, d-tree, and PBT – Power Boosting Tree Classifier. I am applying these algorithms for maintaining more accurate results and making the transactions effective.*

Keywords- Data and web mining, PBT, Credit card transactions, Safety, d-tree.

I. INTRODUCTION

In the way of using data mining, firstly sorting of the data will performed, secondly identifying the patterns and relationships and then after performing data analysis and problem solving operations. System of payment by using credit card is the easiest way of payment and the most common type of transact financially. The frauds in credit card are meant to the unlicensed use of the details of credit cards without the permission of the owner.

The iterative process consists of the following steps:

1) Data cleaning: also known as data cleansing, it is a phase in which noisy data and irrelevant data are removed from the collection.
2) Data integration: In this step, multiple data sources, often heterogeneous, may be combined in a common source.
3) Data selection: data relevant to the analysis is decided on and retrieved from the data collection in this step.
4) Data transformation: also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
5) Data mining: it is the crucial step in which clever techniques are applied to extract patterns potentially useful.
6) Pattern evaluation: in this step, strictly interesting patterns representing knowledge are identified based on given measures.
7) Knowledge representation: is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

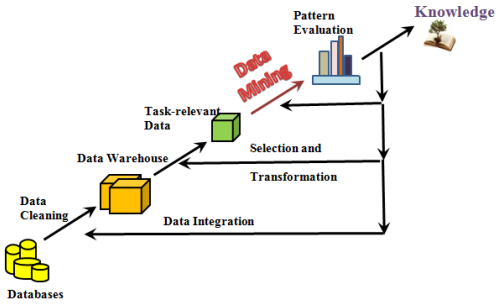

Fig.1 Data Mining as a step in Knowledge Discovery [16]

II. BASIC OVERVIEW OF NAIVE BAYES

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Here,
P(A) => independent probability of A (prior probability)
P(B) => independent probability of B
P(B|A) => conditional probability of B given A (likelihood)
P(A|B) => conditional probability of A Given B (posterior probability)

III. BASIC OVERVIEW OF PBT CLASSIFIER

In predictive data mining or we can say classification and regression power boosting tree technique is generally boosting gradient technique. It is formerly use for one of the range of strategies of boosting.

A linear combination of these trees is then used to build the PBT classifier.

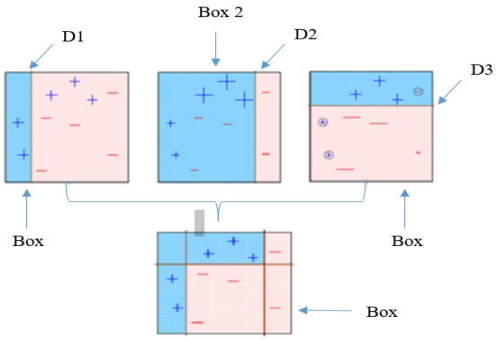

Fig.2Classic Classification Example [12]

IV. LITERATURE SURVEY

An Adaptive Approach on Credit Card Fraud Detection Using Transaction Aggregation and Word Embedding

According to [1], Character-level word embedding and a sliding window-based automated training dataset creation approach are the major focus of this study. To translate the name to a vector of real numbers, character-level word embedding is required, and the merchant's name may be utilized as a unique characteristic to detect fraudulent conduct. To adaptively prevent idea drift, the sliding window-based automatic training dataset generation approach is retrained over time.

Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection

The random under-sampling approach was employed for skewed datasets, and the three different proportions of datasets were used in this study. Logistic Regression, Nave Bayes, and K-Nearest Neighbor are the three machine learning methods employed in this study. The performance of these algorithms is tracked and analyzed to see how well they distinguish and categories fraud and non-fraud transactions in the credit card dataset using the random under sampling technique, and whether or not the performance has improved.

Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison

The process of acceptance or rejection of a transaction occurs in a very short time frame, which can range from micro to milliseconds, and a huge number of related types of transactions occur at the same time. As a result, in order to distinguish between a genuine and a fraudulent transaction, a Fraud Detection Mechanism must be implemented. Accuracy was not employed as a criterion since it is unaffected by skewed data and lacks a decisive response. They used kNN, Naive Bayes, Decision Tree, Logistic Regression, and Random Forest models to predict the likelihood of a fraudulent credit card transaction occurring out of a given number of transactions.

Credit Card Fraud Detection Using Online Boosting with Extremely Fast Decision Tree

According to [4], classifier ensembles have been effectively utilised to improve the performance of single classifiers in data mining and data stream mining. As a result, this work introduces the Online Boosting strategy, which first employs the Extremely Fast Decision Tree as a foundation learner, then assembles them into a single strong online learner, resulting in high prediction success with minimal memory and time costs.

Credit Card Fraud Detection Using Lightgbm Model

According to [5] it is a scalable end-to-end tree boosting strategy that is used by many data scientists to get state-of-the-art outcomes while solving several machine learning issues. Other traditional machine learning models, such as SVM, logistic regression, and Xgboost, were also used in this job. They utilised it to fine-tune parameters such as the learning rate, the number of estimators, the sample rate of rows, the sample rate of columns, the maximum depth of each tree, and the boosting types. Experiments revealed that the lightgbm model beat the other Logistic Regression, SVM, and Xgboost models on both the Auc-Roc score and the Xgboost score.

V. PROPOSED METHOD

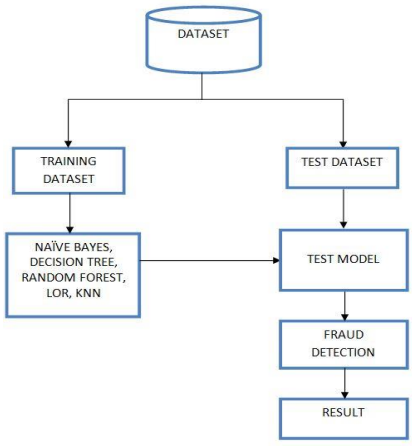

Fig. 3 Flowchart for Existing method [1]

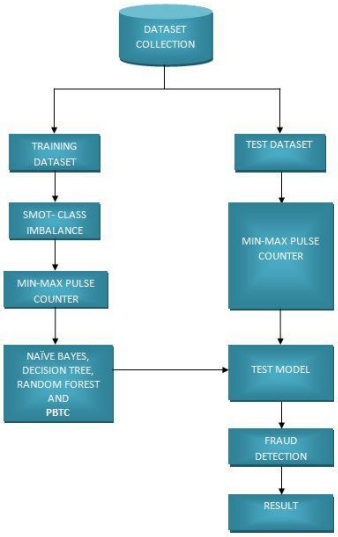

Fig. 4Flowchart for Proposed method

Time	V1	V2	…	V28	Amount	Class
0	-1.35981	-0.07278	…	-0.02105	149.62	0
0	1.191857	0.266151	…	0.014724	2.69	0
1	-1.35835	-1.34016	…	-0.05975	378.66	0
1	-0.96627	-0.18523	…	0.061458	123.5	0
2	-1.15823	0.877737	…	0.215153	69.99	0
. . .	. . .	. . .	. . .	. . .	. . .	. . .
172787	-0.73279	-0.05508	…	-0.05353	24.79	0
172788	1.919565	-0.30125	…	-0.02656	67.88	0
172788	-0.24044	0.530483	…	0.104533	10	0
172792	-0.53341	-0.18973	…	0.013649	217	0

Dataset Description

Table 1 Credit card dataset containing V1 to V28 columns, Time and Amount

Description of Proposed System

- It begins with data collection; in this stage, the input data is gathered in the form of CSV files.
- A method for obtaining context for the incoming data. Preprocessing and cleaning datasets need an understanding of the data. The 'amount' and 'time' columns were not standardized. Principal Component Analysis was used to standardize the remaining data.
- The dataset is then separated into a training dataset and a test dataset, with 80 percent of the data being used to train the model and the remaining 20% being used to test the model, which will be extremely skewed or unbalanced.
- On the dataset, use the SMOTE class imbalance solver approach, which is used to balance class distribution by recreating minority class cases at random.
- On the dataset, there seems to be a fantastic pulse counter. It perpetuates the data by converting the minimum value of each feature to a 0, the positive ranking to a 1, and all other quantities to a decimal between 0 and 1.
- The data is then sent into machine learning methods like Nave Bayes, D-TREE, and Power Boosting Tree (PBT) Classifier once it has been segregated. This stage primarily involves teaching the computer to improve its predicted accuracy by utilising training data.
- Our trained model will be ready for testing once the data has learned enough.
- The learnt model is put to the test with real-world data to see how well it is at predicting the future.
- The model has been implemented after the forecast accuracy reaches the specified level.

Proposed Algorithm

BEGIN
Step 1: Take input from Dataset.
Step 2: Data-preprocessing from Dataset.
Step 3: Divide Training and Testing data from Dataset.
Step 4: Utilize class imbalance solver technique on Dataset
Step 5: Apply Merest – Superlative pulse counter on Dataset
Step 6: Train Model using Naïve Bayes, D-TREE, and Power Boosting Tree (PBT) Classifier algorithm
Step 7: Model Trained
Step 8: Fraud Detection
Step 9: Result
End

Here we had put some snapshots of the implemented algorithms code.

```
import pandas as pd
data=pd.read_csv(r'C:\Users\admin\Desktop\creditcard.csv')
y=data['Class']
data=data.drop(columns=['Class'])
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data,y, test_size=0.20, random_state=42)
```

```
y_train.hist()
```

```
<AxesSubplot:>
```

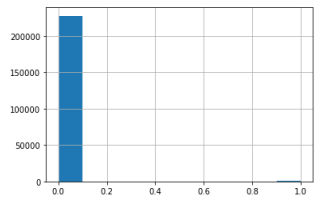

Fig. 5 Code Snippet 1

```
from imblearn.over_sampling import RandomOverSampler, SMOTE
sm = SMOTE(random_state = 42)
X_res, y_res = sm.fit_resample(X_train,y_train)
X_res = pd.DataFrame(X_res)
Y_res = pd.DataFrame(y_res)
Y_res.hist()
```

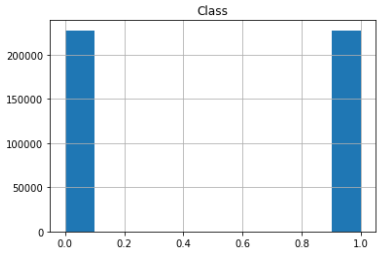

Fig. 6 Code Snippet 2

```
from sklearn.preprocessing import MinMaxScaler as Merest_Superlative_Pulse_Counter
scaler = Merest_Superlative_Pulse_Counter()
X_res=scaler.fit_transform(X_res)
X_res_test=scaler.transform(X_test)
```

Fig. 7 Code Snippet 3

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_res, y_res)
gnb.score(X_res_test,y_test)
```

```
0.9769671008742671
```

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,gnb.predict(X_res_test))
```

```
array([[55565,  1299],
       [   13,    85]], dtype=int64)
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,gnb.predict(X_res_test)))
```

```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99     56864
           1       0.06      0.87      0.11        98

    accuracy                           0.98     56962
   macro avg       0.53      0.92      0.55     56962
weighted avg       1.00      0.98      0.99     56962
```

Fig. 8 Code Snippet 4

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=0,max_depth=1)
clf.fit(X_res, y_res)
clf.score(X_res_test,y_test)
```

```
0.9691899863066605
```

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,clf.predict(X_res_test))
```

```
array([[55122,  1742],
       [   13,    85]], dtype=int64)
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,clf.predict(X_res_test)))
```

```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     56864
           1       0.05      0.87      0.09        98

    accuracy                           0.97     56962
   macro avg       0.52      0.92      0.54     56962
weighted avg       1.00      0.97      0.98     56962
```

Fig. 9 Code Snippet 5

```
from xgboost import XGBClassifier as PBTC #POWER BOOSTING TREE CLASSIFIER
import math
model = PBTC(random_state=0,max_depth=1)
model.fit(X_res, y_res)
float = model.score(X_res_test,y_test)
format_float = "{:.2f}".format(float)
print("PBTC ACCURACY IS:" , format_float)
```

```
[10:58:44] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release
0, the default evaluation metric used with the objective 'binary:logistic'
t eval_metric if you'd like to restore the old behavior.
PBTC ACCURACY IS: 0.99
```

```
from sklearn.metrics import classification_report
model_Pred=model.predict(X_res_test)
modelreport = classification_report(y_test, model_Pred)
print(modelreport)
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     56864
           1       0.13      0.92      0.23        98

    accuracy                           0.99     56962
   macro avg       0.56      0.95      0.61     56962
weighted avg       1.00      0.99      0.99     56962
```

Fig. 10 Code Snippet 6

VI. COMPARISON & RESULT

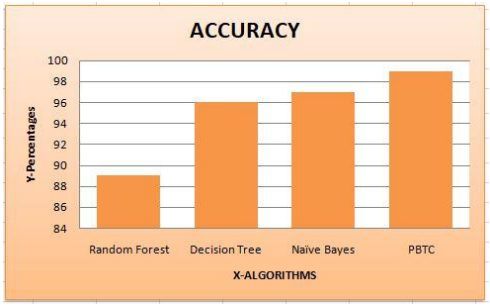

Fig. 11 Comparison of Algorithms Based on Accuracy

Algorithms	Accuracy %
Random Forest	89
Decision Tree	96
Naive Bayes	97
PBTC	99

Table 2 Performance Results

VII. CONCLUSION

The research work was carried out to compare the ability of machine learning algorithms as to how accurately they differentiate and classify the fraud and non-fraud transactions of the credit card dataset with SMOTE technique and to check out if the performance is improved or not. Power Boosting Trees (PBT) showed the optimal performance for all the data proportions as compared to Random Forest (RF), Naive Bayes (NB) and Decision Tree (D-Tree). PBT was successful in getting higher accuracy as compared to Naive Bayes and Decision Tree. The PBT showed the maximum accuracy of 99%, RF showed 89%, NB showed 97.70% and D-Tree showed 96.92%. Also, PBT shows the better Precision, Recall, and F-Measure as compare to RF, NB and D-Tree technique.

REFERENCES

[1] Ali Ye, silkanat(B), Barı, s Bayram, Bilge K¨oro˘glu, and Se, cil Arslan, "An Adaptive Approach on Credit Card Fraud Detection Using Transaction Aggregation and Word Embeddings" © IFIP International Federation for Information Processing 2020, Published by Springer Nature Switzerland AG 2020, I. Maglogiannis et al. (Eds.): AIAI 2020, IFIP AICT 583, pp. 3–14, 2020.

[2] Fayaz Itoo, Meenakshi, Satwinder Singh, "Comparison and analysis of logistic regression, Naı¨ve Bayes and

KNN machine learning algorithms for credit card fraud detection" © Bharati Vidyapeeth's Institute of Computer Applications and Management 2020.

[3] Samidha Khatri, Aishwarya Arora, Arun Prakash Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison" © 2020 IEEE.

[4] Aye Aye Khine, Hint Wint Khin, "Credit Card Fraud Detection Using Online Boosting with Extremely Fast Decision Tree" © 2020 IEEE.

[5] Dingling Ge, Shunyu Chang, "Credit Card Fraud Detection Using Lightgbm Model" © 2020 IEEE.

[6] Altyeb Altaher Taha, Sharaf Jameel Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine " © 2020 IEEE, VOLUME 8, 2020.

[7] J. V. V. Sriram Sasank, G. Ram sahith, K.Abhinav, Meena Belwal, "Credit Card Fraud Detection Using Various Classification and Sampling Techniques: A Comparative Study" Proceedings of the Fourth International Conference on Communication and Electronics Systems (ICCES 2019).

[8] Debachudamani Prusti, Santanu Kumar Rath, "Fraudulent Transaction Detection in Credit Card by Applying Ensemble Machine Learning techniques", 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT 2019).

[9] Kaithekuzhical Leena Kurien, Dr. Ajeet Chikkamannur, "Detection And Prediction Of Credit Card Fraud Transactions" © International Journal of Engineering Sciences & Research Technology (IJESRT 2019).

[10] Rishi Banerjee, Gabriela Bourla, Steven Chen, Mehal Kashyap, Sonia Purohit, Jacob Battipaglia, "Comparative Analysis of Machine Learning Algorithms through Credit Card Fraud Detection" © 2018 IEEE MIT Undergraduate Research Technology Conference (URTC).