Use Machine Learning Techniques to Identify Credit Cards Fraud Detection

Katta Veera Venkata Surya Teja[1]; Kamana Vijay Vamsi[2]
Kunadharaju Vinod Varma[3]; Kandikatla Sandeep[4]
Swarnandhra College of Engineering and Technology

Abstract:- Credit card fraud is an easy target. E-commerce and many other online sites collect money online, increasing the risk of online fraud. As fraud increases, researchers have begun using different learning techniques to detect and analyze fraud in online businesses. The main goal of this article is to design and develop a new streaming data transfer fraud method that aims to identify customer context and extract behavior from the business. Card holders are divided into different groups according to transaction fees. Sliding windows are then used to combine transactions from different cardholder groups, allowing the behavior of each group to be separated. The different groups were then divided into classes for training. Classes with better scores can be selected as the best way to predict fraud. Therefore, the following feedback strategy is adopted to solve the drift law problem. In this article, we use the European credit card fraud dataset.

Keywords:- *Credit Card Fraud Detection; Machine Learning Algorithms; Vague Search; See Instructions; Unsupervised Learning; Kev Faib Algorithm.*

I. INTRODUCTION

Using Machine Learning to Detect Credit Card Fraud Credit card fraud is a huge problem for financial institutions and cardholders. As business volume and complexity continues to increase, traditional fraud detection methods, such as legal systems, become ineffective at detecting fraud over time. As a result, there is a growing need for technology and flexibility in combating fraud. Machine learning (ML) has become a powerful tool for instant fraud detection. Machine learning algorithms can analyze large amounts of transaction data, identify patterns and distinguish legitimate transactions from fraudulent transactions with high accuracy. The process of using machine learning for credit card verification requires several key steps: Data collection: Collect transaction data from a variety of sources, including registration, customer information, and fraud history information. Preliminary data: Cleaning and preparing data for analysis, which may include handling missing values, coding categorical variables, and standardizing numerical properties. Custom options: Identify relevant attributes (such as transaction value, location, time, and cardholder behavior) that help distinguish legitimate transactions from fraud. Model training: Use control learning algorithms (such as logistic regression, decision trees, random forests, support vector machines, or neural networks) to train models on recorded data. This model learns to classify changes as fraudulent or legitimate based on certain characteristics. Model evaluation: Evaluate your model's performance using metrics such as accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) curve Training: Place the sample on your credit card and immediately run it to test incoming emails. goods. Business. The model flags suspicious transactions for further review by fraud analysts or automatically blocks them to prevent financial loss. Continuous evaluation and improvement: Continuously updating and improving machine learning models based on new data and fraud patterns to increase their accuracy and adaptability over time. Credit card identification using machine learning protects financial institutions and cardholders from potential losses and security threats by providing effective ways to detect and reduce low-level fraud.

II. LITERATURE SURVEY

➢ *Summary:*

There is no fixed standard for fraud. They constantly change their behavior; so we have to use unsupervised training. Fraudsters are realizing that new technology allows them to commit fraud through online transactions. While scammers assume user behavior is constant, fraud patterns can change rapidly. Therefore, fraud detection tools need to use unsupervised learning to detect online transactions; because some scammers commit online fraud in the middle of the process and then move on to other processes. The purpose of this paper is 1) to focus on false positive cases based on historical or experimental research and to develop a model combining power absorption autoencoder and Boltzmann limit machine (RBM) with which traditional business processes can be reproduced. Report anomalies. from the original model. Autoencoder (AE)-based deep learning is an unsupervised learning method that uses feedback from the algorithm. RBM has two layers; access layer (visible) and hidden layer. In this work, we use AE, RBM, and H2O for deep learning using Google's Tensorflow library. The results show the squared error, mean square error, and area under the curve.

- The goal of data analysis is to uncover hidden patterns and use them to support informed decision-making in a variety of situations. Nowadays, with the advancement of technology, credit card fraud is increasing day by day and becoming an easy target for fraud. Credit card fraud is a major problem in the financial services industry, costing billions of dollars each year. Detecting fraud is a difficult task due to the difference between private and public information and the lack of information exchange in the real world. In this article, we use various machine learning algorithms to detect credit card fraud using real data. We also use these algorithms to use super classifiers through learning. We've identified the most important changes that will improve the accuracy of credit card fraud detection. We also discuss and compare the performance of various supervised machine learning algorithms available in the literature with the super classifier we use in this paper.

- In 2017, credit card fraud cost North American companies $3 billion. The rise of digital payment methods such as Apple Pay, Android Pay and Venmo means that losses due to fraud will also increase. Deep learning holds great promise in solving credit card fraud by allowing organizations to leverage customer profile history and transaction details collected during operations. A 2017 study found that deep learning provided comparable results to fraud detection methods such as gradient boosted trees and logistic regression. But deep learning has many topologies. In addition, many parameters such as the number of neurons in the hidden layer of the neural network used to create the model can also affect the results. In this paper, we evaluate the performance of a small field of deep learning, ranging from neural networks to physical and memory (e.g., long-term memory) topologies, on approximately 80 million previously flagged credit cards. seen as dishonest and legitimate. We leverage this high-performance, variable cloud environment to solve previous problems such as class ambiguity and scalability. Our evaluation provides general guidelines on criteria for measuring performance. We also conducted a study on how we can improve upon finding the source of credit card fraud; thus allowing financial institutions to reduce fraud prevention costs. 4. Credit card fraud occurs frequently and causes huge financial losses. Criminals may use techniques such as Trojans or phishing to steal others' credit card information. That's why fraud detection is so important because when a thief uses a stolen card to make a purchase, they can instantly detect fraud. One way is to use all historical market data, including traditional market and fraud market, to get good behavior/fraud based on machine learning and bite and then use these features to check if the transaction is fraudulent or not. This article uses two random forests to train normal and abnormal markets. We compare two random forests with different algorithms and analyze their performance in fraud detection. The data used in our experiment belongs to an e-commerce company in China. Index Terms—Random forests, decision trees, credit card fraud.
- Due to the rapid advancement of e-commerce technology, credit card usage has increased significantly. Considering that credit cards are the most common payment method, credit card fraud is on the rise. With the rise of online payments and the shift to Internet transactions in cashless transactions, fraud has become an important element of business security. Credit card technology has become increasingly available and new payment methods have emerged for electronic services such as e-commerce and mobile payments. A safe and reliable fraud prevention system is necessary to prevent credit card fraud. Various methods have been proposed to identify credit cards, including prediction methods and algorithms. These create powerful algorithms that allow data to be classified as normal or suspicious. However, despite the increased use of technology, credit card fraud continues. This work presents a method for detecting credit card fraud using random forests. Use a random forest algorithm to identify profiles and available user data. This approach increases the accuracy of the output before the process of analyzing profiles to detect fraud. Current and future fraud measures are also compared and analyzed. In this article, the random forest classification model is used and the performance of the model is evaluated using the graphical representation of true and false distributions.

III. METHODS

Credit card identification using machine learning involves using a variety of techniques to detect fraud at many legitimate businesses. The following procedure outlines typical steps for using machine learning to detect credit card fraud:

➢ *Data Collection and Progress:*

Obtain transaction history data from credit card companies or financial institutions. This information should include the transaction price, time, location, transaction type (online, in-person), and possibly other relevant information. Process data by handling missing values, removing duplicates, and normalizing numbers.

➢ *Data Analysis (EDA):*

Analyze data to learn about distribution, identify patterns, and understand factors that support fraudulent transactions.

➢ *Feature Engineering:*

Creating relevant features from raw data can help distinguish fraud. Feature engineering may include developing new features, modifying existing features, and planning media releases.

➢ *Data Classification:*

Divide the dataset into training, reference and testing files. The training method is used to train the model, the optimization method is used to tune the hyperparameters, and the testing model is used to evaluate the performance of the model.

➢ *Model Selection:*

Selecting the appropriate machine learning algorithm for fraud detection. Commonly used algorithms include logistic regression, random forests, gradient boosting machines (GBM), support vector machines (SVM), and neural networks. Consider combining multiple models to improve performance.

➢ *Model Training:*

Train the selected model using the training data. During training, the model learns patterns and relationships between features and their corresponding texts (fraud or legal transactions).

➢ *Model Evaluation:*

Use another method to validate the training model. Measurements such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC) are often used to evaluate model performance.

➢ *Hyperparameter Optimization:*

Optimization of hyperparameters using techniques such as grid search, random search, or Bayesian optimization to select alternative models to improve performance.

➢ *Model Deployment:*

Deploy training models to a production environment that can control changes over time. Use analytics to track your model's performance over time and retrain your model regularly to replace bad models.

➢ *Post-Deployment Monitoring:*

Check the performance of the model in the production environment to check if there is any degradation in its performance. appeared. Use feedback to incorporate new information and iterate regularly to keep the model updated.

IV. METHODOLOGY

Random forest is a popular machine learning algorithm and supervised learning process. It can be used in classification and regression problems in machine learning. It is based on the concept of ensemble learning, which is the process of combining multiple classifiers to solve complex problems and improve model performance.

As the name suggests, "Random Forest is a classification system that, given a dataset, will use multiple decision trees across multiple cluster distributions and average them as the accuracy of the dataset." "Random forests are immutable. Relying on the decision tree, a prediction is made from each tree and the final outcome is predicted based on the majority vote of the prediction.

- Trees in the forest are more accurate in terms of outcome, and also the Over-matching problem is avoided.
- Random Forest, predictive analytics It is one of the best learning models for; make it a machine learning task.

➢ *Then*

Random Forest model, Results are combined to predict the model underlying the decision. final Model g, fi , simpler models is the number. Here each base classifier is a simple decision tree. The technique of using multiple models to obtain better predictions is called model integration. In random forest, all underlying models are created independently using different patterns drawn from the data.

The popularity of the random forest algorithm is due to its user-friendliness and flexibility, which allows it to solve the problem of classification and recovery. The power of this algorithm lies in its ability to process complex data and reduce overfitting; this makes it useful for many prediction tasks in machine learning.

One of the most important features of the random forest algorithm is that it can process both continuous (such as regression) and categorical variables (such as distribution). Distribution and recycling are better. In this article, we will learn how random forests work and how to use random forests in job distribution.

➢ *Code:*

- *Input:*

✓ Import Numpy as np
✓ Import Pandas as pd
✓ Ntshuam Matplotlib.Pyplot as plt
✓ Ntshuam Seaborn as sns
✓ Dataset = pd.read_csv('archive (3.zip) ')

➢ *Dataset*

- Output:

Table 1 Dataset

Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.620
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.690
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.660
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.500
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.990
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	......
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348	1.436807	0.250034	0.943651	0.823731	0.770
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226	-0.606624	-0.395255	0.068472	-0.053527	24.790
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134	0.265745	-0.087371	0.004455	-0.026561	67.880
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.123205	-0.569159	0.546668	0.108821	0.104533	10.000
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.008797	-0.473649	-0.818267	-0.002415	0.013649	217.000

284807 Rows × 31 Columns

- *Input:*

X = dataset.iloc[: , :-1].values

y = dataset.iloc[: , -1].values

Imblearn.over_sampling 與 RandomOverSampler

Ros = RandomOverSampler(random_state=0)< br>x_res, y_res = ros.fit_resample(X, y)

Sklearn.model_selection train_test_split

x_train , x_test , y_train , y_test = train_test_split (x_classifier., . RandomForest6nstClassifier, 0. =-1)

Sınıflandırıcı .fit(x_train , y_train)

- *Output:*

RandomForestClassifier(n_estimators=64, n_job=-1, random_state=0)

- *Input:*

Corrmat = dataset.corr()

Plt.figure(figsize=(10,10))

Sns.heatmap(corrmat , vmax=0,8 , square=True)

Plt.show()

- *Output:*

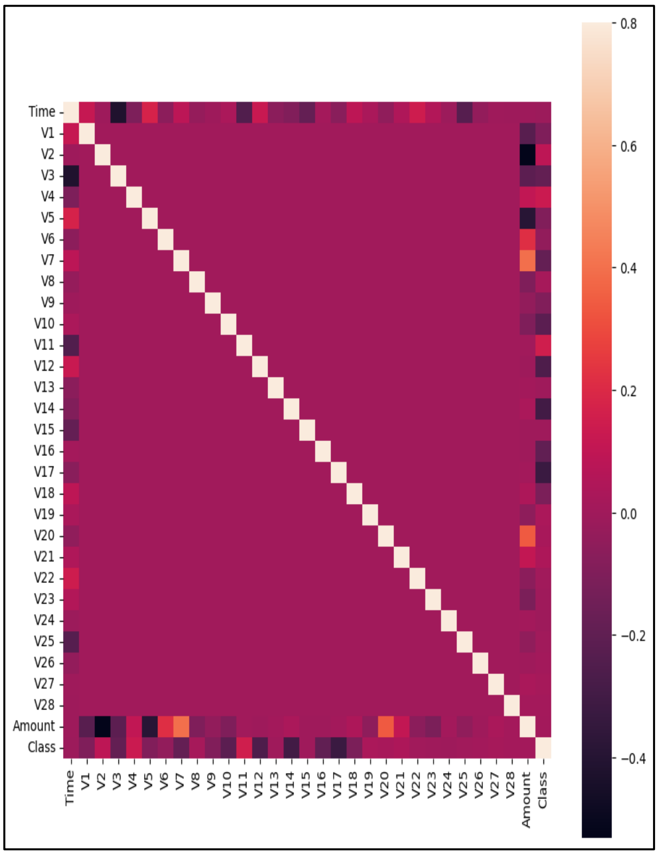

Fig 1 Correlation Plot

- *Input:*

```
y_pred = classifier.predict(x_test)

n_errors = (y_pred != y_test).sum()

n_errors
```

- *Output:*

```
8
```

- *Input:*

```
Sklearn.metrics fusion_matrix,accuracy_score

Cm = fusion_matrix(y_test,y_pred)

Sns.heatmap(cm,annot=True)

Print(accuracy_score(y_test,y_pred)))
```

- *Output:*

```
0.9999296554877513
```

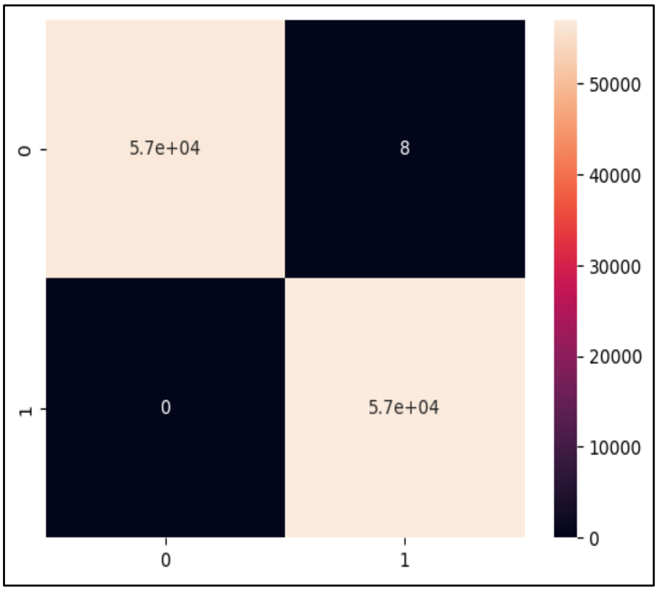

Fig 2 Confusion Matrix

- *Input:*

```
Sklearn.metrics'i  Precision_score

Sensitive_score(y_test , y_pred)
```

- *Output:*

```
0.9998596294216732
```

- *Input:*

```
Sklearn.metrics recall_score

Recall_score（y_test， y_pred）
```

- *Output:*

```
1.0
```

- *Input:*

```
Los ntawm sklearn.metrics import classification report

Print(classification_report(y_test , y_pred))
```

- *Output:*

Precision	Recall	F1-score	Support	
0	1.00	1.00	1.00	56742
1	1.00	1.00	1.00	56984

Accuracy			1.00	113726
Macro Avg	1.00	1.00	1.00	113726
Weighted Avg	1.00	1.00	1.00	113726

V. CONCLUSION

In conclusion, using machine learning for credit card fraud is an important practice that helps financial institutions and businesses protect their customers and assets from fraud. Using advanced algorithms and techniques such as logistic regression, random forests or neural networks, models can be created that can detect fraudulent transactions with high-profile individuals.

REFERENCES

[1]. Pumsirirat, A. and Yan, L. (2018). Credit card identification using deep learning-based autoencoders and constrained Boltzmann machines. International Journal of Advanced Computer Science and Applications, 9(1).

[2]. Muhammad, Emad and Berouz Far. - Supervised machine learning algorithms for detecting fraudulent credit card transactions: a comparative study. - IEEE Annals of Computer History, IEEE, July 1, 2018.

[3]. Roy, Abhimanyu and others. - Deep learning detects fraud in credit card transactions. – 2018 Systems and Information Engineering Design Symposium (SIEDS), 2018.

[4]. Xuan, Shiyang, et al. - Random forest for credit card verification. - 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), 2018.

[5]. Awoyemi, John O., et al. - Detecting credit card fraud using machine learning techniques: a comparative analysis. - 2017 International Conference on Computer Networks and Informatics (ICCNI), 2017.