Engineering and Technology Journal e-ISSN: 2456-3358

Volume 09 Issue 02 February-2024, Page No.-3516-3520

DOI: 10.47191/etj/v9i02.11, I.F. – 8.227

© 2024, ETJ

Analysis of Credit Card Fraud Detection Performance Using Random Forest Classifier & Neural Networks Model

Steven Wijaya[1], Wilfredo Wesly[2], Kristina Ginting[3], Abdi Dharma[4]

[1,2,3,4]Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Prima Indonesia, Jl. Sampul No. 3, Medan, North Sumatra, Indonesia

ABSTRACT: This research discusses credit card fraud detection using machine learning algorithms, specifically Random Forest Classifier and Neural Networks. Research methods include the EDA (Exploratory Data Analysis) stage, data preprocessing, building Random Forest and Neural Networks models, as well as model evaluation. The data used comes from the Kaggle dataset that has been provided. Data analysis is carried out using Pandas to understand the structure and content of the data, while preprocessing involves checking for duplicates, displaying data information, and statistical descriptions. The research results show that the Random Forest model achieved an accuracy of 96% in detecting credit card fraud, while Neural Networks also provided good results. A comparison of the performance of the two algorithms shows that both are effective in detecting fraud. Suggestions for further development include comparing the performance of the model with other algorithms, exploring the factors that influence fraud detection, and developing a more complex and adaptive detection system. The positive implication of the results of this research is increased efficiency in credit card fraud detection, which can provide major benefits in protecting consumers and financial institutions from detrimental fraudulent activities. References used in the research are also included to support the validity and accuracy of the findings obtained.

KEYWORDS: Machine Learning, Credit Card Fraud Detection, Random Forest Classifier, Neural Networks.

1. INTRODUCTION

Credit card fraud is one example of data manipulation in the e-commerce industry. Because credit card fraud is so common, preventing it can be difficult. Therefore, it is important to identify credit card fraud as soon as it occurs. Determining the validity of a transaction is a fraud detection process[1]. Credit/ Debit cards/ any financial services are small plastic cards given to members of certain financial organizations with proper identity and verification[2].

Economic fraud is a hassle that has been confirmed to be a hazard and has some distance-reaching effects within the financial industry[3]. There are several types of financial fraud such as credit card fraud, securities fraud, insurance fraud, etc[4]. Credit card transactions in Indonesia increased to 28.360 in May 2022 from 23.452 in the same month the previous year. Credit cards are often the focus of crimes such as transaction fraud, which is the term for illegal transactions carried out by third parties using the credit card holder's direct data, because of their widespread use[5].

The recognition of credit card cellular payments has provided more possibilities for fraudsters to commit credit card fraud, through strategies together with credit card searches, counterfeit card fraud, payment fraud, etc[6]. Credit card fraud is characterized by an entity gaining unauthorized access to a card to make purchases[7]. Card and debit cards are used for card operations; they are utilized to make purchases of goods and services both online and in physical locations. When making transactions online, fraud is made easier because it is sufficient to have the card data without having to provide them[8].

Using fraudulent credit cards to make purchases without paying for them is known as credit card fraud. This study examines current techniques for detecting credit card fraud and classifies them into two major groups. Furthermore, we look into how Neural Network models can be used to resolve the credit card fraud detection problem [9]. Economic fraud is a main difficulty this is becoming worse and affects the government, corporate community, and financial sector in many ways[10]. Offline payments are distinguished by being physical. For offline payments, the cardholder must be present and provide their PIN. The credit card number of the cardholder is the first thing needed for online payment fraud. Credit cards are used both online and offline[11].

This has a significant impact on increasing online commercial transactions or electronic commerce[12]. Due to rapid technological advances, the use of credit cards for financial activities has increased drastically. Advanced techniques are used in credit card activities, requiring effective technology to detect fraud to secure the payment system[13].

2. RESEARCH METHODS

This research is quantitative research that uses datasets that have been verified and classified as fraudulent or non-fraudulent transactions. This research will limit itself to credit card fraud detection methods that use Neural Networks and Random Forest

Classifier algorithms. Using publicly available credit card transaction data as a starting point, this research will test how well machine learning algorithms overall perform in identifying credit card fraud using Neural Networks and Random Forest Classifier methods.

2.1. Work Procedures

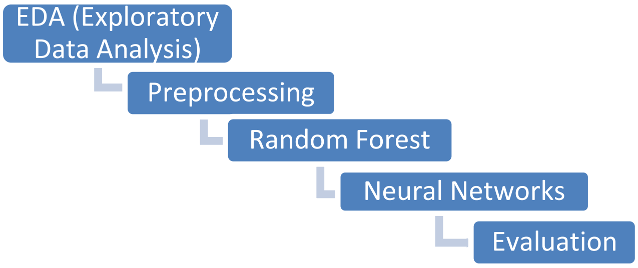


Figure 1. Preparation Phase

A. EDA (Exploratory Data Analysis)

Data analysis is carried out at the EDA stage to identify characteristics and trends in credit card transaction data. This analysis includes distribution variables, correlation between variables, as well as data visualization to find insights that can be used in developing models for credit card fraud detection. EDA is an important first step in understanding the characteristics of data before entering the preprocessing and machine learning model development stages.

B. Preprocessing

Data preprocessing is carried out before building a credit card fraud detection model using machine learning algorithms. Preprocessing steps that can be taken include:

1. Reading the dataset: Collections and data containing verified credit card transactions are loaded into the system for analysis.
2. Data Information: Basic information about the data set, such as number of entries, columns, and data types, is checked.
3. Data Cleaning: Any missing values or inconsistencies in the data set are handled and corrected.
4. Exploratory Data Analysis: Initial insights and patterns in the data are explored to understand its characteristics.
5. Preprocessing: Data is prepared for model building through techniques such as normalization, categorical variable coding, and feature scaling.

C. Random Forest

This step includes building a machine learning model using the Random Forest Classifier algorithm. Separating data as training data and test data, model training, model testing, and model evaluation using metrics such as accuracy, confusion matrix, and classification report to evaluate model performance.

D. Neural Networks

Prepare data that has gone through the data cleaning and data preparation stages. Choosing a neural network model architecture that is appropriate for the problem of credit card fraud detection. To train and test the model, break up the problem of credit card fraud detection. To train and test the model, break up the data into education and test sets. Building neural network models using the TensorFlow library in a Python environment. Train the model using training data and adjust parameters such as the number of layers, number of neural networks, and activation function. Test the model using test data to evaluate credit card fraud.

E. Evaluation

At this stage, the evaluation results are carried out on the machine learning model that has been built using the Random Forest Classifier and Neural Networks algorithms. Evaluation is carried out by separating the data into training and testing sets, training the model, testing the model, and evaluating model performance using metrics such as accuracy, confusion matrix, and classification report. The results of this evaluation are important to determine how well the model can detect credit card fraud based on the available transaction data.

2.2. Tools and Materials

A. Tools

In this research, the tools used have the following specifications: Windows 11 Pro 64 bit, Processor Ryzen 7 4800H with Radeon Graphics 2.90 Ghz – 4.2GHz, ROM 512GB, RAM 16GB, GPU Nvidia Geforce RTX 2060 6GB.

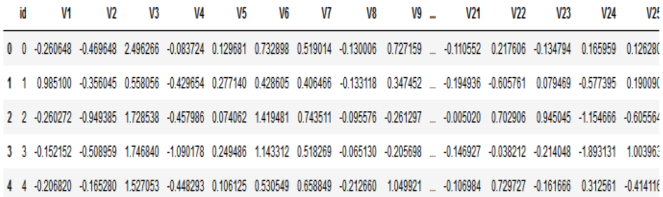
B. Materials

This research uses an online dataset from Kaggle with a link: https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023.

3. RESULTS AND DISCUSSION

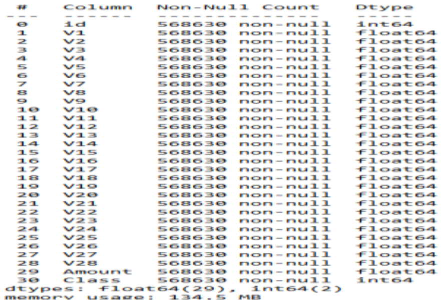
3.1.1.Results EDA (Exploratory Data Analysis)

PD.read_csv('creditcard_2023.csv') is a command in Python using the Panda's library to read and insert data from a CSV (Comma-separated values) file into a df variable as a data frame. df. head() is used to display the first 5 rows of the df data frame.



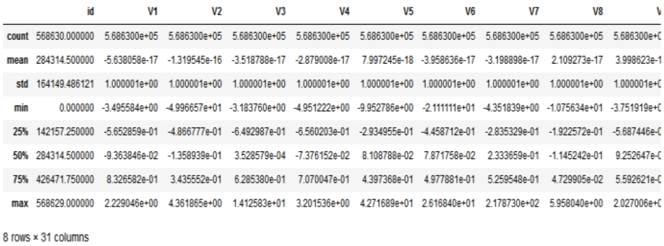
	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25
0	0	-0.260648	-0.469648	2.496266	-0.083724	0.129681	0.732898	0.519014	-0.130006	0.727159	...	-0.110552	0.217606	-0.134794	0.165959	0.12628
1	1	0.985100	-0.356045	0.558056	-0.429654	0.277140	0.428605	0.406466	-0.133118	0.347452	...	-0.194936	-0.605761	0.079469	-0.577395	0.19009
2	2	-0.260272	-0.949385	1.728538	-0.457986	0.074062	1.419481	0.743511	-0.095576	-0.261297	...	-0.005020	0.702906	0.945045	-1.154666	-0.60556
3	3	-0.152152	-0.508959	1.746840	-1.090178	0.249486	1.143312	0.518269	-0.065130	-0.205698	...	-0.146927	-0.038212	-0.214048	-1.893131	1.00396
4	4	-0.206820	-0.165280	1.527053	-0.448293	0.106125	0.530549	0.658849	-0.212660	1.049921	...	-0.106984	0.729727	-0.161666	0.312561	-0.41411

df.info() is a command in Python using the panda's library to display information about a data frame.



```
 #    Column   Non-Null Count    Dtype
---   ------   --------------    -----
 0    id       568630 non-null   int64
 1    V1       568630 non-null   float64
 2    V2       568630 non-null   float64
 3    V3       568630 non-null   float64
 4    V4       568630 non-null   float64
 5    V5       568630 non-null   float64
 6    V6       568630 non-null   float64
 7    V7       568630 non-null   float64
 8    V8       568630 non-null   float64
 9    V9       568630 non-null   float64
 10   V10      568630 non-null   float64
 11   V11      568630 non-null   float64
 12   V12      568630 non-null   float64
 13   V13      568630 non-null   float64
 14   V14      568630 non-null   float64
 15   V15      568630 non-null   float64
 16   V16      568630 non-null   float64
 17   V17      568630 non-null   float64
 18   V18      568630 non-null   float64
 19   V19      568630 non-null   float64
 20   V20      568630 non-null   float64
 21   V21      568630 non-null   float64
 22   V22      568630 non-null   float64
 23   V23      568630 non-null   float64
 24   V24      568630 non-null   float64
 25   V25      568630 non-null   float64
 26   V26      568630 non-null   float64
 27   V27      568630 non-null   float64
 28   V28      568630 non-null   float64
 29   Amount   568630 non-null   float64
 30   Class    568630 non-null   int64
dtypes: float64(29), int64(2)
memory usage: 134.5 MB
```

df. describe() is a command in Python using the Panda's library to display a statistical description of a data frame.



	id	V1	V2	V3	V4	V5	V6	V7	V8	V
count	568630.000000	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+05	5.686300e+0
mean	284314.500000	-5.638058e-17	-1.319545e-16	-3.518788e-17	-2.879008e-17	7.997245e-18	-3.958636e-17	-3.198898e-17	2.109273e-17	3.998623e-1
std	164149.486121	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+00	1.000001e+0
min	0.000000	-3.495584e+00	-4.996657e+01	-3.183760e+00	-4.951222e+00	-9.952786e+00	-2.111111e+01	-4.351839e+00	-1.075634e+01	-3.751919e+0
25%	142157.250000	-5.652859e-01	-4.866777e-01	-6.492987e-01	-6.560203e-01	-2.934955e-01	-4.458712e-01	-2.835329e-01	-1.922572e-01	-5.687446e-0
50%	284314.500000	-9.363846e-02	-1.358939e-01	3.528579e-04	-7.376152e-02	8.108788e-02	7.871758e-02	2.333659e-01	-1.145242e-01	9.252647e-0
75%	426471.750000	8.326582e-01	3.435552e-01	6.285380e-01	7.070047e-01	4.397368e-01	4.977881e-01	5.259548e-01	4.729905e-02	5.592621e-0
max	568629.000000	2.229046e+00	4.361865e+00	1.412583e+01	3.201536e+00	4.271689e+01	2.616840e+01	2.178730e+02	5.958040e+00	2.027006e+0

8 rows × 31 columns

This diagram displays the distribution of data consisting of three features, namely "Density", "Fraud", and several other features, which have not been written clearly. "Density" data is located on the horizontal axis, while "Fraud" is located on the vertical axis.

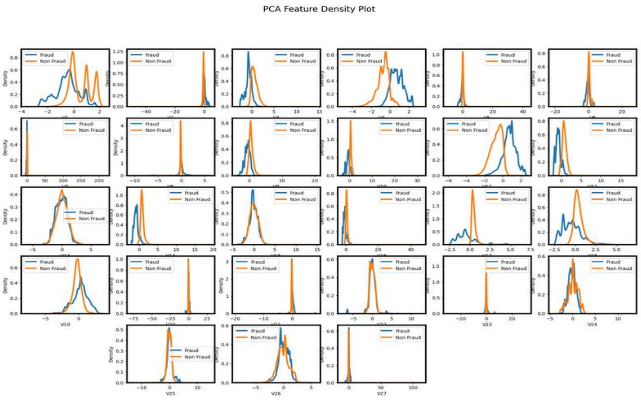


Figure 2. EDA (Exploratory Data Analysis)

3.1.2.Preprocessing

The following are the stages and results of the preprocessing carried out:

1. Read the dataset using ‘pd.read_csv()’ and display the header with ‘df.head()’.
2. Display data information using 'df.info()', which shows that the dataset consists of 568,630 rows and 31 columns, with mixed data types.
3. Display a description of the data using df. describe(), which shows general statistics such as mean, standard deviation, and quartiles.
4. Check whether there are duplicate values using df. duplicated().any(), which indicates that there are no duplicate values.
5. Perform EDA (Exploratory Data Analysis) using several visualizations, such as pie charts and density plots.
6. Preprocessing: Defining X and the target value (Y) using df. drop() to remove the 'id' and 'Class' columns from the dataset, Performing feature scaling using StandardScaler from sklearn. preprocessing.

3.1.3.Modeling with Random Forest

Random Forest is an ensemble learning method that utilizes many decision trees (tree-1, tree-2, ..., tree-n) as classifiers. Each tree will produce a certain classification (Class-X or Class-Y). To get the final result, a majority vote will be carried out on the classification of each tree. For example in the image, tree-1 and tree-n produce Class-X classification, while tree-2 produces Class-Y classification. By voting with a majority, the final result is Class-X.

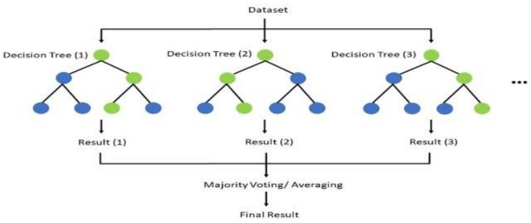
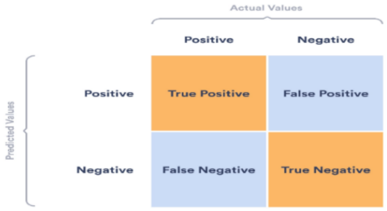


Figure 3. Modeling with Random Forest

A. Confusion Matrix

For the performance of the classification model, an image called a confusion matrix is used. It compares predicted values (Positive and Negative) using actual values (Positive and Negative) and groups them into four types.

- True Positive (TP): The model correctly predicts a positive outcome.
- False Positive (FP): The model incorrectly predicts a positive outcome.
- True Negative (TN): The model correctly predicts negative outcomes.
- False Negative (FN): The model incorrectly predicts a negative outcome.



	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Figure 4. Confusion Matrix

B. Accuracy

Accuracy is the comparison of the number of correct cases predicted (TP+TN) with the total number of cases, namely correct cases (TP+TN) plus incorrect cases (FP+FN).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

C. Precision

Precision is the ratio of the number of cases correctly classified as positive (TP) to the total number of cases predicted to be positive (TP+FP).

$$Precision = \frac{TP}{TP + FP}$$

D. Recall

Recall is the ratio of the number of cases correctly classified as positive (TP) to the total number of positive cases (TP+FN).

$$Recall = \frac{TP}{TP + FN}$$

E. F1-Score

F1-Score is a measure of classification performance that takes both precision and recall values in its calculations, taking into account the harmonic average of these two values using the equation F1-score = 1 (2 x Precision x Recall / (Precision + Recall)).

$$F1\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.1.4.Modeling with Neural Networks

The results of Neural Networks training consist of two main values, namely loss and val_loss. Loss is the error or gap between the desired value and the value produced by the model, while val_loss is an error in data validation. These values are used to measure model performance and perform optimization. Apart from that, Neural Networks will also produce various internal parameters, such as weights and biases, which are used in the calculation process.

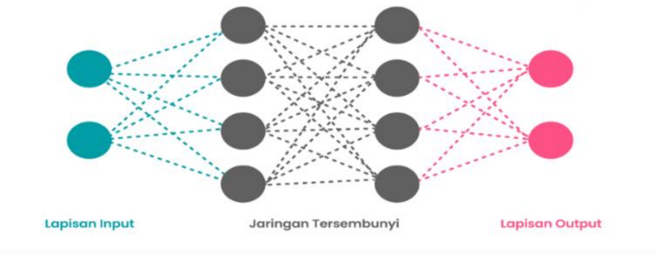
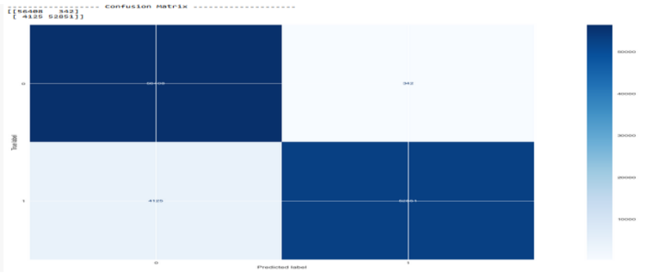


Figure 5. Modeling with Neural Networks

3.1.5.Evaluation result

A. Confusion Matrix Random Forest

The first row represents the number of actual positive samples (56408) and the number of false positives (342). False positives are events where the algorithm incorrectly classifies a negative sample as a positive sample. The second row represents the number of false negatives (4125) and the number of actual negative samples (52851).

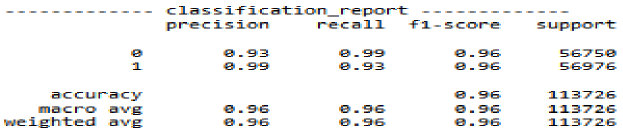


```
------------------- Confusion Matrix -------------------
[[56408   342]
 [ 4125 52851]]
```

Figure 6. Confusion Matrix Random Forest

B. Classification Report Random Forest

The table shows the classification report for the binary classification model with classes ‘0’ and ‘1’. The report includes precision, recall, and F1-score for each class, as well as support. Weighted average takes into account the number of instances for each class, whereas macro average treats all classes equally. Macro average precision was 0.99, recall 0.96, and F1-score 0.96.

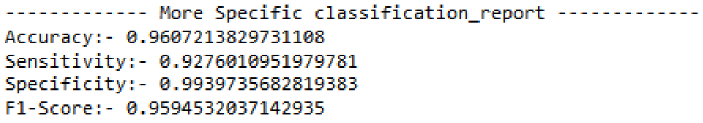


```
------------- classification_report -------------
              precision    recall  f1-score   support

           0       0.93      0.99      0.96     56750
           1       0.99      0.93      0.96     56976

    accuracy                           0.96    113726
   macro avg       0.96      0.96      0.96    113726
weighted avg       0.96      0.96      0.96    113726
```

Figure 7. Classification Report Random Forest

C. More Specific Classification Report Random Forest

A more specific classification report shows the accuracy, sensitivity, specificity, and F1-Score of a classification model. Accuracy shows the percentage of correct model predictions, namely 96.07%. Sensitivity or recall shows the model's ability to detect the positive class, namely 92.76%. Specificity shows the model's ability to detect the negative class, namely 99.39%. F1-Score calculates the harmonic performance of sensitivity and precision, which is 95.94%. However, there is an error in using the negative sign (-) in the F1-Score. F1-Score must be positive and lower than 1.0. So, the correct F1-Score is 0.9594532037142935.



```
------------- More Specific classification_report -------------
Accuracy:- 0.9607213829731108
Sensitivity:- 0.9276010951979781
Specificity:- 0.9939735682819383
F1-Score:- 0.9594532037142935
```

Figure 8. More Specific Classification Report Random Forest

D. Receiver Operating Characteristic Random Forest

The ROC curve is a graph used to show the performance of binary classification, where the x-axis shows the average false positives and the y-axis shows the average true positives. Therefore, the ROC area of 0.99 indicates that this classification model is almost perfect for classifying the data.

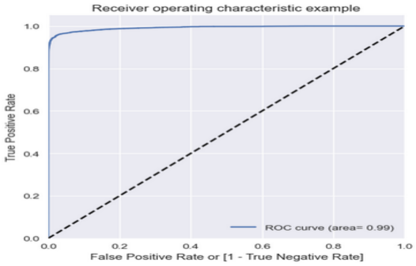


Figure 9. Receiver Operating Characteristic Random Forest

E. Neural Networks

The results of Neural Networks training consist of two main values, namely loss and val_loss. Loss is the error or gap between the desired value and the value produced by the model, while val_loss is an error in data validation. These values are used to measure model performance and perform optimization. Apart from that, Neural Networks will also produce various internal parameters, such as weights and biases, which are used in the calculation process.

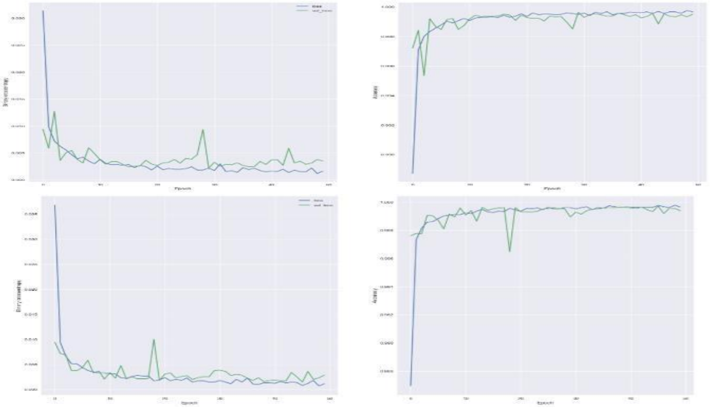


Figure 10. Neural Networks

3.2. Discussion

3.2.1. Performance of the Random Forest Classifier Algorithm in Credit Card Fraud Detection

The Random Forest Classifier algorithm shows excellent performance in detecting credit card fraud. In this research, the algorithm succeeded in achieving 96% accuracy in predicting fraudulent transactions from all transactions. Apart from that, the evaluation results show that this algorithm has a precision of 93%, a recall of 99%, and an F1-Score of 96% for the fraudulent transaction class (class 0). Meanwhile, for the non-fraudulent transaction class (class 1), this algorithm has a precision of 99%, recall of 93%, and F1-Score of 96%. This shows that the Random Forest Classifier algorithm is very effective in identifying fraudulent transactions with a high level of accuracy [10].

3.2.2. Performance of Neural Network Algorithms in Credit Card Fraud Detection

In this study, the performance of the Neural Network algorithm in detecting credit card fraud was evaluated. The research results show that NN can detect credit card fraud with a high level of accuracy, so it can be an effective solution in preventing losses due to credit card fraud[9].

3.2.3. Comparison of Algorithm Performance Between Random Forest Classifier and Neural Networks

In this research, the Random Forest Classifier algorithm succeeded in achieving 96% accuracy in predicting fraudulent transactions, with high precision, recall, and F1-Score values for both transaction classes [10]. In comparison, previous research using Neural Networks also achieved good results in detecting credit card fraud with high accuracy. However, it should be noted that evaluating the performance of Neural network models on validation datasets is also important to assess the generalization ability of the model[9].

3.2.4. Implications of Research Results

These results show significant improvements compared to previously used traditional machine learning models. With increased detection efficiency, both financial institutions and individuals can more effectively address the problem of credit card fraud, reducing losses caused by fraudulent activities [17].

4. CONCLUSION

This research discusses credit card fraud detection using machine learning algorithms, specifically Random Forest Classifiers and Neural Networks. A comparison of the performance of the two algorithms shows that both are effective in detecting fraud. The results of this research have positive implications for improving the efficiency of credit card fraud detection, as well as contributing to the understanding of the application of machine learning algorithms in the problem of credit card fraud detection.

REFERENCES

1. Pangestu, G. T., & Rosyda, M. (2022). Sentiment Analysis Tweet Pilkada 2020 Saat Pandemik COVID-19 di Media Sosial Twitter Menggunakan Metode 1D Convolutional Neural Network. JURNAL MEDIA INFORMATIKA BUDIDARMA, 6(2), 1017.
2. Kumar Joshi, A., Shirol, V., Jogar, S., Naik, P., & Yaligar, A. (2020). Credit Card Fraud Detection Using Machine Learning Techniques.
3. Bagga, S., Goyal, A., Gupta, N., & Goyal, A. (2020). Credit Card Fraud Detection using Pipeling and Ensemble Learning. Procedia Computer Science, 173, 104–112.
4. Khan, S., Alourani, A., Mishra, B., Ali, A., & Kamal, M. (2022). Developing a Credit Card Fraud Detection Model using Machine Learning Approaches. International Journal of Advanced Computer Science and Applications, 13(3), 411–418.
5. Ningsih, P. T. S., Gusvarizon, M., & Hermawan, R. (2022). Analisis Sistem Pendeteksi Penipuan Transaksi Kartu Kredit dengan Algoritma Machine Learning. Jurnal Teknologi Informatika Dan Komputer, 8(2), 386–401.
6. Jiang, S., Dong, R., Wang, J., & Xia, M. (2023). Credit Card Fraud Detection Based on Unsupervised Attentional Anomaly Detection Network. Systems, 11(6).
7. Bach Nguyen, V., Ghosh Dastidar, K., Granitzer, M., & Siblini Worldline, W. (2022). The Importance of Future Information in Credit Card Fraud Detection (Vol. 151).
8. Mijwil, M. M., & Salem, I. E. (2020). Credit Card Fraud Detection in Payment Using Machine Learning Classifiers. Asian Journal of Computer and Information Systems, 8(4).
9. Al Balawi, Salwa, and Nojood Aljohani. "Credit-card fraud detection system using neural networks." Int. Arab J. Inf. Technol. 20.2 (2023): 234-241.
10. Aburbeian, AlsharifHasan Mohamad, and Huthaifa I. Ashqar. "Credit Card Fraud Detection Using Enhanced Random Forest Classifier for Imbalanced Data." International Conference on Advances in Computing Research. Cham: Springer Nature Switzerland,2023.
11. Karthikeyan, T., M. Govindarajan, and V. Vijayakumar. "An effective fraud detection using competitive swarm optimization based deep neural network." Measurement: Sensors 27 (2023): 100793.
12. Dewi, N. K. A., & Mahyuni, L. P. (2020). Pemetaan bentuk dan pencegahan penipuan e-commerce. E-Jurnal Ekonomi Dan Bisnis Universitas Udayana, 9, 851-878.
13. Beigi, S., & Amin-Naseri, M.-R. (2020). Credit Card Fraud Detection using Data mining and Statistical Methods. Journal of AI and Data Mining, 8(2), 149–160.
14. LUO, X., WANG, S., CHEN, H., & LUO, Z. (2023). The Utility Impact of Differential Privacy on Credit Card Data in Machine Learning Algorithms. Procedia Computer Science, 221, 664–672.
15. Gupta, P., Varshney, A., Khan, M. R., Ahmed, R., Shuaib,M., & Alam, S. (2022). Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques. Procedia Computer Science, 218, 2575–2584.
16. Rtayli, N., & Enneya, N. (2020). Selection features and support vector machine for credit card risk identification. Procedia Manufacturing, 46, 941–948.
17. Zhang, Y. F., Lu, H. L., Lin, H. F., Qiao, X. C., & Zheng, H. (2022). The Optimized Anomaly Detection Models Based on an Approach of Dealing with Imbalanced Dataset for Credit Card Fraud Detection. Mobile Information Systems, 2022.