**JURNAL TEKNOLOGI DAN OPEN SOURCE**
Vol. 7, No. 2, December 2024, pp. 167~177
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 153/E/KPT/2023
DOI: 10.36378/jtos.v7i2.3890

# Application of Deep Learning Algorithm to Detect Fraud in Online Transaction Networks

**Ridwan Dwi Irawan[1*], Agus Fatkhurohman[2]**
[1,2,3]Fakultas Ilmu Komputer, Universitas Duta Bangsa Surakarta, Surakarta, Indonesia
[1,2,3]Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia

**Article Info**

***Article history:***

Received 11 20, 2024
Revised 11 26, 2024
Accepted 12 25, 2024

***Keywords:***

Deep Learning
Convolutional Neural Networks
Long Short-Term Memory
Fraud

**ABSTRACT**

Online transaction fraud is a severe problem that may cost businesses and people a lot of money. This paper suggests using deep learning algorithms to detect fraud as a remedy to this issue. These algorithms were chosen based on their ability to handle large amounts of intricate data and identify patterns that are difficult to identify using traditional techniques. Important components of this research include gathering and preprocessing transaction data, creating deep learning models, and assessing model performance. This investigation examines a variety of financial transaction types that may have involved fraud. The deep learning approach uses deep neural network designs, including Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), to maximize detection accuracy. The study's findings demonstrate that the deep learning models created are excellent at identifying questionable transactions and can lower the false positive rate, which raises the overall effectiveness of fraud detection systems. As a result, deep learning algorithms have demonstrated a high degree of efficacy in identifying fraudulent activity inside internet-based transaction networks, so they play a vital role in fraud prevention.



***Corresponding Author:***

Ridwan Dwi Irawan
Fakultas Ilmu Komputer
Universitas Duta Bangsa Surakarta
Surakarta, Indonesia
Email: ridwan.irawan@udb.ac.id


## 1. Introduction

Fraud in internet-based transaction networks has become a major threat to the security of financial systems worldwide. Fraud is increasing along with the growth of e-commerce and digital financial services. This fraud can not only cost businesses and consumers a lot of money, but it can also destroy public trust in digital payment systems, which are increasingly popular in society[1]. Therefore, a fast and efficient fraud detection system is needed to find and stop suspicious actions. The use of machine learning algorithms, especially deep learning, has been proven to have the capacity to increase accuracy and efficiency in fraud detection and is one of the innovative approaches that is starting to be widely used[2].

Conventional methods for detecting fraud traditionally include heuristic-based rules and statistical analysis. However, these methods increasingly show limitations as the volume and complexity of transaction data increases. Conventional methods are often unresponsive to identifying complex and hidden fraudulent modus operandi found in highly volatile transaction data[3]. Machine learning, particularly deep learning, has

a better chance in this regard because of its ability to process and understand complex patterns with a precision far beyond traditional methods [4], [5].

Through the use of multi-layered artificial neural networks, deep learning, a branch of machine learning, enables more sophisticated data analysis. The system can learn complex and abstract representations of data through these networks, which is very useful for identifying anomalies in financial transactions. This method allows deep learning-based fraud detection systems to find fraud patterns that are often missed by conventional methods. This method makes detection more accurate and responsive [6], [7]. To identify suspicious transaction patterns, architectures such as Convolutional Neural Networks (CNN) are very effective[8].

Previous studies have shown that CNN algorithms detect suspicious transactions more accurately than heuristic-based methods. CNNs are designed to extract spatial features from data, such as transaction matrices, which help find anomalous patterns that form over time [9], [10]. Another study found that CNN's ability to map frequently recurring patterns in credit card fraud and other financial transactions makes it a highly effective tool for detecting fraud[11].

In the initial stage of this method, financial transaction data is collected from various sources to identify the diversity of transaction patterns. Once collected, this data then goes through a pre-processing process, which includes cleaning, normalization, and transformation. This process is very important to reduce noise in the data and ensure the quality and consistency of transaction data that will be used in the model [12]. Without this process, the model may fail to capture relevant patterns or may get caught in unsuspecting anomalies. Therefore, data pre-processing is the basis for a more comprehensive transaction analysis [13].

The CNN model was developed and trained using prepared data after the data was pre-processed. To avoid overfitting, which can cause the model to be inaccurate when applied to new data, the transaction data was divided into training and testing data during the training stage. This process is intended to make the model more accurate in detecting fraud [14].

Then, the trained model is evaluated to ensure optimal detection performance using metrics such as accuracy, precision, recall, and F1 score. This evaluation is important to measure the model's ability to detect fraud accurately so that it can prevent errors in finding suspicious transactions. In addition to being able to detect fraud with high accuracy, an effective detection system also has a low false positive rate[15], [16]. This is very important because high false positives can reduce system efficiency and disrupt user experience.

Applying deep learning models to fraud detection systems has been shown to reduce the number of false positives generated, which is a major problem in conventional fraud detection methods. Thus, this technology can detect fraudulent activities more accurately, maintain the security of financial transactions, and at the same time reduce the possibility of false detection [17].

This study shows that not only is it necessary to improve accuracy, but also to develop a system that is able to handle the ever-evolving changes in fraud patterns. To maintain the performance of the model in the long term, continuous monitoring and updating are needed.

Long Short-Term Memory (LSTM) is a very powerful variant of Recurrent Neural Networks (RNN), specifically designed to address the vanishing gradient problem and is capable of capturing long-term dependencies in sequential data. Lindemann et al.[18] presents a comprehensive survey highlighting the capabilities of LSTM in time series prediction, demonstrating its flexibility across a wide range of applications. The regularization mechanism, as empirically evaluated by Chung et al. [19], becomes a key element in LSTM, allowing the selection of information to be remembered or forgotten. This structure has also proven effective in machine translation tasks, as discussed by Cho et al.. [20], where the RNN-based encoder-decoder architecture leverages LSTM to generate high-quality phrase representations. In addition, the application of a two-stage attention mechanism, as explored by Qin et al. [22], have significantly improved the performance of LSTM in time series forecasting, demonstrating its adaptability and reliability. This progress, supported by extensive exploration of neural network architectures [23], continues to make LSTM a critical element in sequence modeling and predictive analytics.

## 2. Research Methods

This research begins with the stage of collecting and pre-processing financial transaction data, which is an important step to ensure the quality of the data used in model development. The collected transaction data must go through a cleaning process to remove anomalies or irrelevant data, normalization to equalize the data scale, and transformation to prepare the data to fit the format that can be used by the deep learning model.

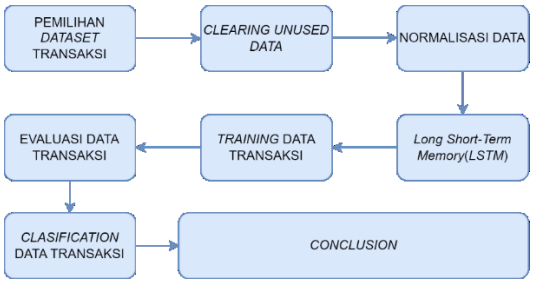


Figure1. General Flow of Research

Once the data is ready, the development of a deep learning model is carried out using architectures such as Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN). LSTM is chosen because of its ability to handle sequential data, which is important for understanding continuous patterns in financial transactions, while CNN is known to be effective in capturing feature patterns from complex data. The model is trained using data divided into training sets and test sets. This division is important to prevent overfitting, where the model becomes too specific to the training data and loses generalization to new data. With this technique, the model is expected to be able to recognize fraud patterns more effectively and accurately. After the model is trained, the next stage is performance evaluation. The evaluation metrics used include accuracy, precision, recall, and F1 score.

**Data Understanding**

Kaggle's "Credit Card Fraud Detection" dataset contains credit card transactions suspected of being fraudulent. Here is a description of the datase.

Table 1. Dataset Understanding

| | | |
|---|---|---|
| Objective | | Detect credit card transactions that are likely fraudulent. |
| **Feature** | **Time** | Time in seconds from first transaction |
| | **V1 - V28** | 28 anonymous features generated through PCA (Principal Component Analysis) to protect personal information. These features cover various attributes related to transactions. |
| | **Amount** | The amount of money involved in the transaction |
| | **Class** | A label indicating whether the transaction is fraudulent (1) or not (0). |
| **Additional information** | **Data Quantity** | This dataset consists of 284,807 transactions, of which a small portion (around 0.172%) are fraudulent transactions. |
| | **Class Imbalance** | This dataset has a significant imbalance between normal and fraud classes, which may affect the results of the prediction model. |
| | **Use** | This dataset is often used for the development and evaluation of fraud detection models using machine learning techniques. |

This dataset contains 284,807 credit card transactions, with only about 0.172% of the total transactions categorized as fraudulent, creating a significant class imbalance to maintain user data privacy. The last feature, Class, is a label indicating whether the transaction is fraudulent (1) or not (0). This dataset is often used by researchers and practitioners in finance to explore better fraud detection techniques.

The dataset is highly imbalanced, with fraudulent transactions accounting for only 0.17% of the total. This reflects a real-world scenario where fraudulent transactions are rare, making this dataset an excellent source for studying imbalance handling techniques such as undersampling, oversampling, and cost-sensitive learning. Insights gained from this dataset can inform real-world systems for fraud detection in financial institutions, online marketplaces, and e-commerce platforms, contributing to reducing financial losses and increasing transaction security.

## Data Preparation

After reduction from 284,807 transactions to 11,528 transactions, the resulting dataset is more concise but still representative for credit card fraud analysis purposes.

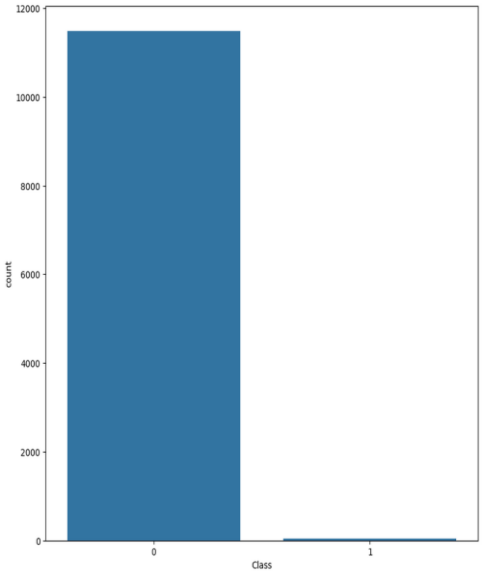


Figure2. Separation of Fraud and Non-Fraud-Based Transaction Data

In the latest subset of the credit card fraud detection dataset, there are a total of 11,479 transactions, comprising 11,430 non-fraudulent transactions (Class 0) and 49 fraudulent transactions (Class 1). This distribution highlights a significant class imbalance, with fraudulent transactions representing only about 0.43% of the total data. This mirrors a real-world challenge in fraud detection, where the vast majority of transactions are legitimate, and only a small fraction are fraudulent. Although the number of fraudulent transactions is much smaller, preserving this imbalance is crucial to ensure that the developed model accurately reflects real-world scenarios and can effectively handle rare events without biasing predictions toward the majority class.

The class imbalance in this dataset can significantly impact the performance of fraud detection models. Models trained on such data may be prone to overlooking fraudulent transactions, as they are much less frequent compared to valid transactions. Therefore, evaluating the model using appropriate metrics, such as precision, recall, and F1-score, is essential to ensure that the model not only prioritizes non-fraudulent transactions but also accurately identifies fraudulent cases.

Table 2. Data Preparation

| *Class* | Amount of Data |
|---|---|
| 0 | 11479 |
| 1 | 49 |

Based on the available data, there are two classes with significantly different amounts of data: Class 0 contains 11,479 instances, while Class 1 has only 49 instances. This stark difference highlights a pronounced imbalance in class distribution, commonly referred to as class imbalance.

Model Design

The model architecture in this credit card fraud detection research is specifically designed to address the challenges posed by class imbalance while ensuring optimal detection accuracy.

This research architecture for credit card fraud detection is designed to address imbalanced datasets and ensure the model can effectively predict fraudulent transactions. The process begins with data preparation, where irrelevant data is removed to maintain high-quality inputs. Next, the dataset is split into training and testing subsets in an 80:20 ratio. The training data is used to train the model, while the testing data evaluates its performance. The three classification models implemented in this architecture, namely Random Forest, Linear Regression, and K-Nearest Neighbors, are selected because of their ability to handle different data characteristics, such as complexity, linearity, and proximity relationships between features. In the final stage, the model is tested and the results are thoroughly analyzed using various evaluation metrics to ensure that the built model can detect fraud with high accuracy and good generalization to new data. This

architecture is designed with a systematic approach, ensuring that each stage of the research process is well integrated to achieve the goal of more optimal fraud detection.

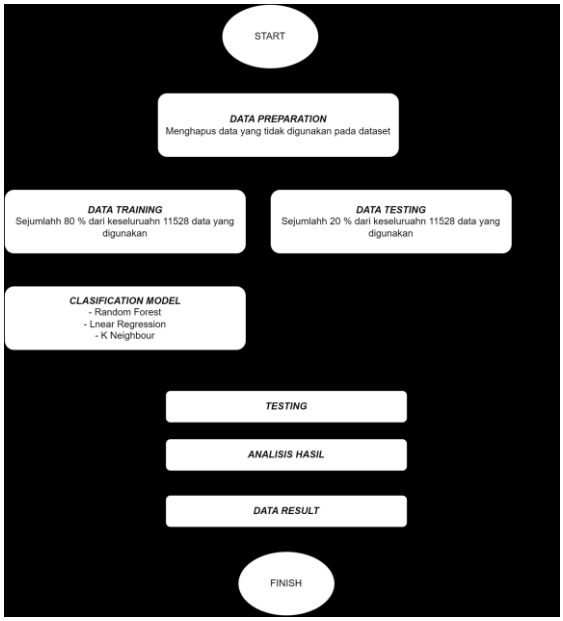


Figure3. Research Architecture

Figure 3 shows a workflow or flowchart of a classification model development process for credit card fraud detection. The process begins with the Data Preparation stage, where unused data in the dataset is removed to ensure that only relevant information is fed into the model. This stage is crucial, as the quality of the data directly impacts the performance of the predictive model. Following data preparation, the dataset is divided into two main subsets: training data and testing data. Specifically, 80% of the total data is allocated for training the model, while the remaining 20% is reserved for testing. This split aligns with standard machine learning practices, where the training data is used to develop the predictive model, and the testing data is employed to evaluate its performance. This approach helps ensure that the model is not overfitting and can generalize well to new, unseen data. Next, three types of classification models are applied to predict whether a transaction is fraudulent or not. The models used are Random Forest, Linear Regression, and K-Nearest Neighbors (KNN). Each model has its own strengths and weaknesses in handling imbalanced data, making them suitable for different aspects of the task.

The process concludes with the Analysis Results and Data Result stages. In these stages, the trained models are evaluated on the test data, their results are analyzed, and their performance in fraud detection is presented. Key evaluation metrics such as precision, recall, and F1-score are used to ensure that the models can accurately and effectively identify fraudulent transactions, even in the context of imbalanced datasets.

## 3. Results and Discussion

The discussion of the research findings and testing results is presented through both theoretical analysis and quantitative descriptions.

### 3.1. *Correlation of Forecasting Algorithm with Classification Model*

The Long Short-Term Memory (LSTM) algorithm in deep learning is very effective when used on data that has a temporal or sequential relationship, such as credit card transaction data from fraud detection datasets. LSTM works by retaining information over a longer period of time, making this model suitable for detecting transaction patterns over time. In the context of credit card fraud detection, Long Short-Term Memory (LSTM) networks can effectively track changes in user behavior by analyzing transaction sequences using time-related features, which may reveal anomalous or fraudulent patterns. On the other hand, Convolutional Neural Networks (CNNs), traditionally used for spatial pattern recognition, can also be adapted for time-series data by transforming transaction data into a matrix form or leveraging techniques

such as 1D convolution. Additionally, traditional algorithms like Random Forest, K-Nearest Neighbors (KNN), and Linear Regression are generally better suited for simpler, more straightforward models compared to the more complex architectures of LSTMs and CNNs.

Machine learning models play a crucial role in credit card fraud detection by providing fast and accurate solutions for classifying transactions as fraudulent or legitimate. Models like Random Forest are particularly effective in handling complex and imbalanced datasets, as they employ an ensemble approach, constructing multiple decision trees to deliver more stable results while being resistant to overfitting. K-Nearest Neighbors (KNN), on the other hand, classifies transactions based on feature similarity, enabling the model to identify patterns in transactions that resemble previous fraudulent cases.

### 3.2. Model Implementation Results

Based on the modeling results presented in the figure, the credit card fraud detection process, using various classification algorithms such as Random Forest, Linear Regression, and K-Nearest Neighbors (KNN), follows systematic stages in data division and model evaluation. The data is split into training data (80%) and testing data (20%), allowing the model to learn to recognize patterns of fraudulent transactions from the training set and then be tested on new, unseen data to assess its performance. The results of model testing will be analyzed to evaluate key metrics such as accuracy, precision, recall, and F1-score, helping to determine how effectively the model detects fraudulent transactions, particularly in the context of significant class imbalance. This evaluation is crucial to ensure that the model delivers accurate results and generalizes well to real-time data, making it suitable for integration into broader fraud detection systems in real-world applications.

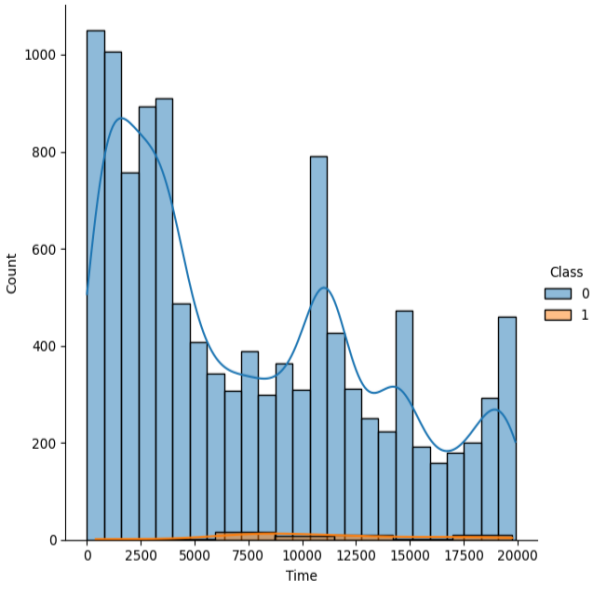


Figure4. Comparison Chart

The figure illustrates the data distribution based on the Time variable for two classes: class 0 and class 1. It is evident that class 0 (represented by blue) contains a significantly larger volume of data compared to class 1 (represented by orange), which aligns with the earlier analysis of class imbalance.

The histogram reveals that the data for class 0 is distributed with a fluctuating pattern over a specific time span. At the beginning of the time span (around 0-2500), the number of class 0 data points reaches its highest peak, with a substantial volume of data. Subsequently, there is a significant fluctuation, with noticeable decreases and increases in the data count until around the 20,000 time mark. This pattern highlights the variation in the number of class 0 data points over the given time interval.

Table 3. Random Forest Algoritma

| Variable | Precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 3444 |
| 1 | 1 | 0,93 | 0,97 | 15 |
| accuracy | | | | 3459 |
| macro avg | 1 | 0,97 | 0,98 | 3459 |
| weighted | 1 | 1 | 1 | 3459 |

| | | | | |
|---|---|---|---|---|
| avg | | | | |
| AUC-ROC | | | | |
| F1-Score | | | 0,965517 | |
| Accuracy | | | 0,999711 | |

Table 3 presents the classification report, showing the performance of the classification model for two classes: class 0 and class 1. Class 0 contains a significantly larger amount of data (3444 instances) compared to class 1 (15 instances). The model demonstrates excellent performance, with precision, recall, and F1-score values close to 1 (indicating near-perfect results) for both classes. For class 0, both precision and recall reach 1.00, meaning the model correctly predicts all class 0 data without error, which is also reflected in the F1-score of 1.00. For class 1, precision remains at 1.00, indicating that all class 1 predictions are correct. However, the recall value of 0.93 suggests that the model missed about 7% of the actual class 1 data, resulting in an F1-score of 0.97 for this class.

The overall model accuracy is 1.00, or nearly perfect (99.97%), indicating that the model correctly predicts almost all data. The average (macro avg) precision, recall, and F1-score are also very high, at 1.00, 0.97, and 0.98, respectively, reflecting consistent performance across both classes. However, the weighted average (based on the number of data points in each class) is close to the class metric value of 0 due to the significant imbalance in the number of data between the classes.

The AUC-ROC value of 0.9666 demonstrates the model's excellent ability to distinguish between class 0 and class 1, with a value approaching 1 indicating near-perfect performance. Overall, the global F1-score of 0.9655 and near-perfect accuracy reflect a highly effective model, although the class imbalance should be considered when interpreting the results.

Table 4. Linear Regression

| Variable | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1 | 0,99 | 0,99 | 3444 |
| 1 | 0,25 | 0,93 | 0,4 | 15 |
| accuracy | | | | 3459 |
| macro avg | 0,63 | 0,96 | 0,7 | 3459 |
| weighted avg | 1 | 0,99 | 0,99 | 3459 |
| AUC-ROC | | | | |
| F1-Score | | | 0,4 | |
| Accuracy | | | 0,987858 | |

The classification report results indicate that the model performs very well in predicting class 0, with precision and recall reaching 0.99 and 1.00, respectively. However, for class 1, the precision is low (0.25), although the recall is relatively high (0.93). As a result, the F1-score for class 1 is only 0.40, suggesting that the model struggles to correctly predict the minority class. While the overall accuracy of the model is high at 98.78%, this is largely driven by the dominance of class 0 data. The AUC-ROC value of 0.9607 demonstrates the model's ability to effectively distinguish between classes, but the low precision for class 1 highlights the need for further improvement.

Table 5. K Nearest Neighbour

| Variable | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1 | 0,94 | 0,97 | 3444 |
| 1 | 0,06 | 0,87 | 0,12 | 15 |
| accuracy | | | | 3459 |
| macro avg | 0,53 | 0,9 | 0,54 | 3459 |
| weighted avg | 1 | 0,94 | 0,97 | 3459 |
| AUC-ROC | | | | |
| F1-Score | | | 0,115556 | |
| Accuracy | | | 0,942469 | |

This classification report highlights the imbalance in model performance across the two classes. Class 0 shows high precision and recall of 1.00 and 0.94, respectively, resulting in a strong F1-score of 0.97. This

indicates that the model accurately classifies the majority class (class 0). However, the performance on class 1 is significantly lower, with a precision of only 0.06 and an F1-score of 0.12, despite a fairly high recall of 0.87. The low precision suggests that the model struggles to correctly predict class 1, leading to a high number of false positives.

Overall, the model accuracy reached 0.94 (94.25%), but this was dominated by the large amount of class 0 data, which was much larger than class 1. The AUC-ROC value of 0.90 indicated that the model's ability to distinguish between the two classes was still good, but the low precision in class 1 indicated the need for improvement.

Table 6. Comparison of 3 Classification Methods

| Variable | precision | recall | f1-score |
|---|---|---|---|
| Precision (Class 0) | 1 | 1 | 1 |
| Precision (Class 1) | 1 | 0,25 | 0,06 |
| Recall (Class 0) | 1 | 0,99 | 0,94 |
| Recall (Class 1) | 0,93 | 0,93 | 0,87 |
| F1-Score (Class 0) | 1 | 0,99 | 0,97 |
| F1-Score (Class 1) | 0,97 | 0,4 | 0,12 |
| Accuracy | 0,9997 | 0,9879 | 0,9425 |
| Macro Avg Precision | 1 | 0,63 | 0,53 |
| Macro Avg Recall | 0,97 | 0,96 | 0,9 |
| Macro Avg F1-Score | 0,98 | 0,7 | 0,54 |
| Weighted Avg Precision | 1 | 1 | 1 |
| Weighted Avg Recall | 1 | 0,99 | 0,94 |
| Weighted Avg F1-Score | 1 | 0,99 | 0,97 |
| AUC-ROC | 0,9667 | 0,9607 | 0,9047 |
| Global F1-Score | 0,9655 | 0,4 | 0,1156 |

From the comparison of the three classification reports above, it is evident that the model performs exceptionally well in the first figure, where the precision, recall, and F1-score for both classes are very high (approaching 1.00), with an accuracy of 99.97% and an AUC-ROC of 0.9667. In the second figure, model performance decreases in the minority class (class 1), with precision dropping to 0.25 and the F1-score to 0.40. Although recall remains high at 0.93, the low precision leads to a slight decrease in both the F1-score and AUC-ROC (0.9607). In the third figure, there is a more significant decline in performance for class 1, with a very low precision of 0.06, resulting in an F1-score of 0.12 for class 1. The overall accuracy drops to 94.25%, and the AUC-ROC decreases further to 0.9047.

In conclusion, the model demonstrates its best performance in the first table, with a decline in the second and third figures. The decrease in performance is particularly evident in the minority class, where precision and F1-score are significantly impacted. This suggests that the model struggles with class imbalance, with the majority class heavily influencing the predictions. Therefore, implementing strategies such as data balancing is crucial to enhance the model's ability to predict the minority class and improve overall classification performance.

## 4. Conclusion

The conclusion of this study shows that the implemented classification model has varying performance in detecting two unbalanced classes, with class 0 as the majority class and class 1 as the minority class. The results of the analysis of three classification scenarios show that the precision, recall, and F1-score for the minority class (class 1) experience a significant decrease when the amount of data in that class is much less. In the first scenario, the model has optimal performance with an AUC-ROC of 0.9667 and a high F1-score for both classes. However, in the second and third scenarios, performance decreases, especially in the minority class, where low precision (0.25 and 0.06 respectively) causes the overall accuracy and discrimination ability of the model to decrease.

By incorporating the Long Short-Term Memory (LSTM) approach, the model is expected to enhance its ability to capture complex patterns and address data imbalance, particularly in cases involving sequential or time-series data. LSTM is well-known for its ability to tackle imbalanced data issues by learning temporal context and efficiently processing sequences. The use of LSTM can improve both recall and precision in the minority class, leading to an overall boost in model performance. Furthermore, combining LSTM with data balancing techniques, such as oversampling or undersampling, offers an effective solution for correcting class imbalance. This approach can ultimately improve the F1-score and AUC-ROC, ensuring a more reliable and accurate classification model.

***Acknowledgments***

The author would like to express sincere gratitude to Universitas Duta Bangsa Surakarta for its financial support in the implementation of this research. The assistance provided by the university played a crucial role in ensuring the smooth progress and success of this study. It is hoped that the results of this research will make a meaningful contribution to the advancement of knowledge and provide benefits to various stakeholders.

## References

[1] F. Zamachsari, N. P.-S. dan T. Informasi), and undefined 2021, "Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik," *jurnal.iaii.or.idF Zamachsari, N PuspitasariJurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 2021•jurnal.iaii.or.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://jurnal.iaii.or.id/index.php/RESTI/article/view/2952

[2] E. Waty, I. Sulistiana, E. Siskawati, L. Judijanto, and I. Maghfur, *AKUNTANSI DIGITAL: Transformasi pembukuan di era revolusi industri 4.0 menuju society 5.0*. 2023. Accessed: Nov. 11, 2024. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=b43pEAAAQBAJ&oi=fnd&pg=PA16&dq=+Penerapan+LSTM+dalam+Mendeteksi+Penipuan+Transaksi+Perbankan.+Jurnal+Teknik+Informatika+dan+Sistem+Informasi,+8(2),+34-48.&ots=gyid7wL8TO&sig=QNOoA7WHAzVs4-sEuc4SBrC8vNw

[3] E. S. Budi, A. N. Chan, P. P. Alda, and Muh. A. F. Idris, "Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network," *Resolusi : Rekayasa Teknik Informatika dan Informasi*, vol. 4, no. 5, pp. 502–509, May 2024, doi: 10.30865/RESOLUSI.V4I5.1892.

[4] R. P.-J. D. Data and undefined 2024, "PENERAPAN MACHINE LEARNING DALAM DETEKSI KECURANGAN PADA TRANSAKSI KEUANGAN ONLINE," *pustakailmu.idRA PutraJurnal Dunia Data, 2024•pustakailmu.id*, Accessed: Nov. 11, 2024. [Online]. Available: http://www.pustakailmu.id/index.php/duniadata/article/view/87

[5] B. Tarissa, T. D.-D. J. of Accounting, and undefined 2024, "PENERAPAN MACHINE LEARNING DAN DEEP LEARNING PADA PENINGKATAN DETEKSI CREDIT CARD FRAUD-A SYSTEMATIC LITERATURE REVIEW," *ejournal3.undip.ac.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://ejournal3.undip.ac.id/index.php/accounting/article/view/46092

[6] L. Mahya, T. Tarjo, … Z. S.-J. R. A., and undefined 2023, "Intelligent Automation Of Fraud Detection And Investigation: A Bibliometric Analysis Approach," *ejournal.umm.ac.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://ejournal.umm.ac.id/index.php/jrak/article/view/28487

[7] F. Zamachsari and N. Puspitasari, "Penerapan Deep Learning dalam Deteksi Penipuan Transaksi Keuangan Secara Elektronik," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 203–212, Apr. 2021, doi: 10.29207/RESTI.V5I2.2952.

[8] R. P.-J. D. Data and undefined 2024, "PENERAPAN MACHINE LEARNING DALAM DETEKSI KECURANGAN PADA TRANSAKSI KEUANGAN ONLINE," *pustakailmu.id*, Accessed: Nov. 11, 2024. [Online]. Available: http://www.pustakailmu.id/index.php/duniadata/article/view/87

[9] S. Prasetyo, T. D.-D. J. of Accounting, and undefined 2024, "Penerapan Machine Learning, Deep Learning, Dan Data Mining Dalam Deteksi Kecurangan Laporan Keuangan-A Systematic Literature Review," *ejournal3.undip.ac.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://ejournal3.undip.ac.id/index.php/accounting/article/view/46146

[10] U. Kolbia, N. D.-J. I. Informatika, and undefined 2024, "ANALISIS KECURANGAN DALAM MENGHADAPI PENIPUAN DI SITUS E-COMMERCE MENGGUNAKAN RANDOM FOREST; PENDEKATAN MACHINE LEARNING," *ejournal.gunadarma.ac.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://ejournal.gunadarma.ac.id/index.php/infokom/article/view/11787

[11] S. S. SupriyantoN. & P. D. K. Nurmalitasari, *LQ45 stock price predictions using the deep learning method*, vol. Int J Adv Res Publ. 2020.

[12] A. Wibiyanto, A. W.-S. S. K. dan, and undefined 2023, "Penerapan Algoritma Multiclass Support Vector Machine dan TF-IDF Untuk Klasifikasi Topik Tugas Akhir," *jom.fti.budiluhur.ac.idADD Wibiyanto, A WibowoSKANIKA: Sistem Komputer dan Teknik Informatika, 2023•jom.fti.budiluhur.ac.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://jom.fti.budiluhur.ac.id/index.php/SKANIKA/article/view/2999

[13] A. Zuhairah, "Penerapan Algoritma Random Forest, Support Vector Machines (Svm) dan Gradient Boosted Tree (Gbt) Untuk Deteksi Penipuan (Fraud Detection) Pada Transaksi," 2022, Accessed: Nov. 11, 2024. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/70536

[14] A. Kurniawan, Y. Y.- Kilat, and undefined 2021, "Pendugaan Fraud Detection pada kartu kredit dengan Machine Learning," *researchgate.net*, Accessed: Nov. 11, 2024. [Online]. Available: https://www.researchgate.net/profile/Yulianingsih-Yulianingsih/publication/356776897_Pendugaan_Fraud_Detection_pada_kartu_kredit_dengan_Machine_Learning/links/61fa9791aad5781d41c81474/Pendugaan-Fraud-Detection-pada-kartu-kredit-dengan-Machine-Learning.pdf

[15] D. Pamungkas, … I. K.-… : J. S., and undefined 2023, "Implementation of Deep Neural Network in the Design of Ethereum Blockchain Scam Token Detection Applications," *sistemasi.ftik.unisi.ac.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/3162/0

[16] Y. Prabowo, … A. N.-S. S., and undefined 2023, "Uji Akurasi Modul KWH Meter Digital PZEM-004T Berbasis Pengendali Digital ESP32," *jom.fti.budiluhur.ac.idY Prabowo, A Narendro, TW Wisjhnuadji, S SiswantoSKANIKA: Sistem Komputer dan Teknik Informatika, 2023•jom.fti.budiluhur.ac.id*, Accessed: Nov. 11, 2024. [Online]. Available: https://jom.fti.budiluhur.ac.id/index.php/SKANIKA/article/view/3064

[17] M. Al Mustofa, … N. N.-B. S. I., and undefined 2024, "Pengembangan Model Rekomendasi Produk UMKM Albis Menggunakan Item Based Collaborative Filtering," *103.133.36.110*, Accessed: Nov. 11, 2024. [Online]. Available: http://103.133.36.110/index.php/BUSITI/article/view/2271

[18] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, pp. 650–655, Jan. 2021, doi: 10.1016/J.PROCIR.2021.03.088.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014, Accessed: Nov. 21, 2024. [Online]. Available: http://arxiv.org/abs/1412.3555

[20] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734, 2014, doi: 10.3115/v1/d14-1179.

[21] D. Hsu, "Multi-period Time Series Modeling with Sparsity via Bayesian Variational Inference," Jul. 2017, Accessed: Nov. 21, 2024. [Online]. Available: http://arxiv.org/abs/1707.00666

[22] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 0, pp. 2627–2633, 2017, doi: 10.24963/ijcai.2017/366.

[23] D. Britz, A. Goldie, M. T. Luong, and Q. V. Le, "Massive exploration of neural machine translation architectures," *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 1442–1451, 2017, doi: 10.18653/v1/d17-1151.

[24] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction," Jan. 2018, Accessed: Nov. 21, 2024. [Online]. Available: http://arxiv.org/abs/1801.02143

[25] R. J. Williams and D. Zipser, “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks,” *Neural Comput*, vol. 1, no. 2, pp. 270–280, Jun. 1989, doi: 10.1162/NECO.1989.1.2.270.
[26] I. Danihelka, G. Wayne, B. Una, N. Kalchbrenner, and A. Graves, “Associative long short-term memory,” *33rd International Conference on Machine Learning, ICML 2016*, vol. 4, pp. 2929–2938, 2016.