Unsupervised Deep Learning for Sequential Anomaly Detection in Financial Transactions Using LSTM and CNN Autoencoders

MS Hasibuan[1,*], Ruki Rizal Nul Fikri[2]

[1,2]Institute Informatics and Business Darmajaya, Indonesia, Jln ZA Pagar Alam 93 A, Bandar Lampung and 35136, Indonesia

ABSTRACT

Anomaly detection in banking transactions is a critical task, particularly in scenarios where labeled fraud data is unavailable. This study proposes an unsupervised deep learning approach for detecting anomalous transaction patterns using sequential modeling with LSTM and CNN autoencoders. The transaction data is preprocessed and transformed into fixed-length sequences to capture temporal and behavioral patterns of individual accounts. Both models are trained to learn normal transaction behavior and identify anomalies based on reconstruction error. Experimental results show that the CNN autoencoder achieves lower reconstruction loss and detects a higher number of anomalies, indicating strong capability in capturing local patterns, while the LSTM autoencoder demonstrates a more selective detection behavior by modeling temporal dependencies. The comparison between the two models reveals partial agreement, where consistently detected anomalies represent high-confidence deviations from normal patterns. Statistical analysis further confirms that anomalous sequences exhibit significantly higher reconstruction errors than normal sequences. The findings highlight the effectiveness of sequence-based unsupervised deep learning methods for anomaly detection in financial transaction data and suggest that combining multiple architectures can improve detection robustness in real-world applications.

Keywords Anomaly Detection, Deep Learning, LSTM Autoencoder, CNN Autoencoder, Financial Transaction Analysis

INTRODUCTION

The rapid advancement of digital banking systems and online financial services has led to a significant increase in the volume and complexity of transaction data. Financial institutions process millions of transactions daily, making it increasingly challenging to monitor and detect suspicious activities in real time. As digital transactions grow, so does the risk of fraud, unauthorized access, and anomalous behavior. Detecting such irregularities is crucial for maintaining financial security, preventing economic losses, and ensuring customer trust. However, the dynamic and evolving nature of fraudulent behavior makes anomaly detection a complex and critical problem in modern financial systems [1].

Traditional fraud detection approaches are predominantly based on supervised learning methods that rely on labeled datasets. These methods require historical data with clearly defined fraud labels to train classification models. In practice, obtaining high-quality labeled data is difficult due to privacy concerns, labeling costs, and the rarity of fraud cases compared to normal transactions. This results in highly imbalanced datasets that can degrade model performance and limit generalization. Consequently, unsupervised anomaly detection

Submitted 25 March 2026
Accepted 10 June 2026
Published 27 October 2026

Corresponding author
MS Hasibuan,
msaid@darmajaya.ac.id

Additional Information and Declarations can be found on page 321

DOI: 10.63913/ftij.v2i4.44

Copyright
2026 Hasibuan and Fikri

Distributed under
Creative Commons CC-BY 4.0

How to cite this article: M. S. Hasibuan, R. R. N. Fikri, "Unsupervised Deep Learning for Sequential Anomaly Detection in Financial Transactions Using LSTM and CNN Autoencoders," *FinTech Innov. J.*, vol. 2, no. 4, pp. 309-323, 2026.

techniques have emerged as a promising alternative, as they do not require labeled data and instead focus on learning normal behavior patterns to identify deviations [2].

In recent years, deep learning-based approaches have become the state of the art in anomaly detection, particularly through the use of autoencoder architectures. Autoencoders are neural networks designed to learn compressed representations of input data and reconstruct it with minimal error. When trained on normal data, these models can effectively identify anomalies based on high reconstruction errors. Among these approaches, Long Short-Term Memory (LSTM) autoencoders have demonstrated strong performance in modeling sequential and temporal data due to their ability to capture long-term dependencies. On the other hand, Convolutional Neural Network (CNN) autoencoders have shown effectiveness in extracting local patterns and structural features, even in sequential contexts. These methods have been successfully applied in various domains, including time-series analysis, network intrusion detection, and financial anomaly detection [3], [4].

Despite these advancements, several research gaps remain. Many existing studies focus on either LSTM-based or CNN-based models independently, without providing a direct comparison of their performance on the same sequential transaction data. In addition, a large portion of prior work treats transaction data as static tabular data, ignoring the sequential nature of user behavior over time. This limits the ability of models to capture temporal patterns that are essential for identifying complex anomalies. Furthermore, there is limited exploration of agreement analysis between different models, which can provide insights into the reliability and consistency of anomaly detection results. These limitations highlight the need for a more comprehensive framework that leverages multiple deep learning architectures and evaluates their complementary strengths [5].

To address these challenges, this study proposes an unsupervised deep learning approach for anomaly detection in bank transactions using both LSTM and CNN autoencoders. The transaction data is transformed into sequential representations based on account-level activity, enabling the models to capture behavioral patterns over time. The proposed framework not only compares the performance of LSTM and CNN models but also analyzes their agreement in detecting anomalies to identify high-confidence irregularities. The main contributions of this study include the application of sequence-based modeling to unlabeled financial data, a comparative evaluation of two state-of-the-art deep learning architectures, and an agreement-based analysis to enhance detection reliability. The findings of this study provide valuable insights into the effectiveness of deep learning techniques for anomaly detection in real-world financial systems.

Literature Review and Related Works

Anomaly detection has become an essential area of research in financial systems due to the increasing complexity of transaction data and the growing risk of fraudulent activities. Traditional approaches for anomaly detection are commonly based on statistical methods and machine learning techniques such as clustering and distance-based models. These methods aim to identify deviations from normal behavior patterns but often struggle to capture complex,

high-dimensional relationships in transaction data [6], [7]. As financial datasets become larger and more dynamic, these limitations have led to the adoption of more advanced techniques.

With the advancement of machine learning, supervised learning approaches have been widely applied for fraud detection tasks. These methods treat anomaly detection as a classification problem and rely on labeled datasets to distinguish between normal and fraudulent transactions. While such approaches have demonstrated high accuracy in controlled environments, their performance is heavily dependent on the availability of high-quality labeled data. In real-world financial systems, labeled fraud data is often limited and highly imbalanced, which reduces model generalization and leads to biased predictions [8], [9]. This challenge has motivated the exploration of unsupervised learning methods that do not require labeled data.

Unsupervised anomaly detection methods focus on modeling normal behavior and identifying deviations without relying on predefined labels. Among these methods, autoencoder-based approaches have gained significant attention due to their ability to learn compact representations of data. Autoencoders reconstruct input data and use reconstruction error as an indicator of anomaly, where higher errors correspond to abnormal patterns. Several studies have demonstrated the effectiveness of autoencoders in detecting anomalies across various domains, including financial transactions, network security, and industrial monitoring [10], [11], [12].

In the context of sequential data, deep learning models such as Long Short-Term Memory networks have been widely adopted due to their ability to capture temporal dependencies. LSTM-based autoencoders are particularly effective in modeling time-series data and have been successfully applied for anomaly detection in sequential environments [13], [14]. These models can learn long-term dependencies and detect irregular temporal patterns that are difficult to capture using traditional methods. However, LSTM models are computationally intensive and may struggle with capturing localized patterns within short sequences.

Convolutional Neural Networks have also been adapted for anomaly detection tasks, especially in scenarios where local patterns and structural features are important. CNN-based autoencoders utilize convolutional layers to extract meaningful representations and have shown strong performance in capturing local dependencies within sequential and structured data [15], [16]. Compared to LSTM models, CNN-based approaches are generally faster to train and more efficient in handling large datasets. However, they may be less effective in modeling long-term temporal relationships.

Several recent studies have explored hybrid and comparative approaches by combining multiple deep learning architectures to improve anomaly detection performance. These approaches aim to leverage the strengths of different models to achieve more robust and accurate detection results [17], [18]. In addition, ensemble-based methods have been proposed to reduce false positives and improve detection reliability by aggregating predictions from multiple models [19]. Despite these advancements, limited research has focused on systematically comparing LSTM and CNN autoencoders within the same framework for sequential financial transaction data.

Furthermore, most existing studies treat transaction data as static tabular data rather than modeling it as a sequence of events. This simplification limits the ability to capture behavioral patterns over time, which are essential for identifying complex anomalies in financial systems [20]. Additionally, few studies have investigated the agreement between different anomaly detection models as a measure of confidence, which can provide valuable insights into the robustness of detected anomalies.

Based on these observations, there is a clear need for a comprehensive approach that integrates sequence-based modeling with multiple deep learning architectures while also analyzing their agreement in detecting anomalies. This study addresses these gaps by proposing a framework that utilizes both LSTM and CNN autoencoders for sequential anomaly detection in bank transactions and evaluates their complementary strengths.

Methodology

Overview of Research Framework

The overall research framework adopted in this study is illustrated in figure 1, which outlines the sequential steps of the proposed anomaly detection approach. The process begins with data collection and preprocessing, followed by feature engineering and transformation into sequential data. The constructed sequences are then used to train two deep learning models, namely the LSTM autoencoder and the CNN autoencoder. After training, reconstruction errors are computed for each sequence and used as anomaly scores. Finally, anomaly detection is performed using statistical thresholds, and the results are analyzed through model comparison and agreement evaluation. This structured pipeline ensures a systematic approach to capturing transaction behavior and identifying anomalous patterns in financial data.

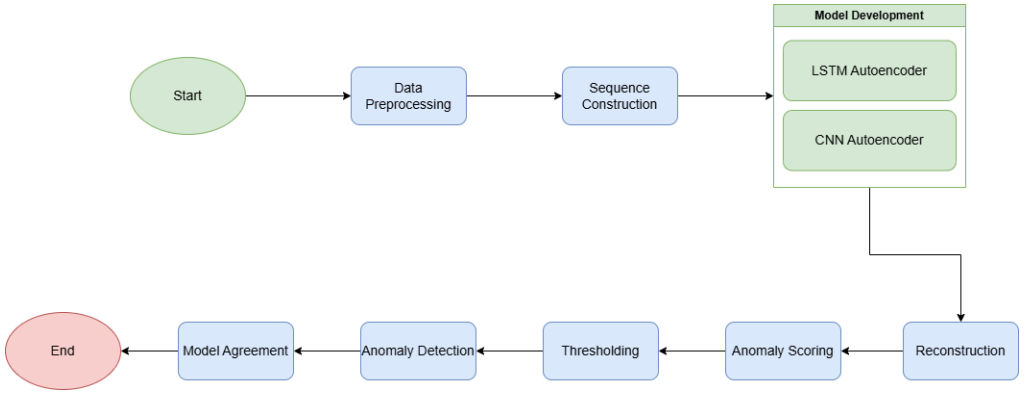


Figure 1 Research Framework

Data Preprocessing

The dataset consists of bank transaction records containing both numerical and categorical attributes. Several preprocessing steps were applied to ensure data quality and relevance. Non-informative attributes such as transaction identifiers, device identifiers, and IP addresses were removed. Temporal information from the transaction timestamp was extracted to generate features such as hour, day of the week, month, and day. A binary indicator representing weekend transactions was also included.

To capture temporal behavior, the time difference between consecutive transactions for each account was calculated as:

$$\Delta t_i = \frac{t_i - t_{i-1}}{3600} \quad (1)$$

$t_i$represents the timestamp of the current transaction and t_{i-1}represents the previous transaction timestamp in seconds. This value is converted into hours to provide a consistent temporal scale.

Numerical features were normalized using z-score standardization:

$$x' = \frac{x - \mu}{\sigma} \quad (2)$$

$\mu$and σrepresent the mean and standard deviation of the feature, respectively. Categorical features were transformed using one-hot encoding, resulting in a high-dimensional feature space.

Sequence Construction

To model transaction behavior as a temporal process, the dataset was transformed into sequences based on account identifiers. Transactions were sorted chronologically and grouped per account. A sliding window approach was used to construct sequences of fixed length T, where each sequence is defined as:

$$X^{(i)} = \{x_i, x_{i+1}, \ldots, x_{i+T-1}\} \quad (3)$$

In this study, the sequence length is set to T=3. Each sequence is represented as a tensor of size TF, where $F$is the number of features.

Model Architecture

The LSTM autoencoder is designed to capture temporal dependencies in sequential data. Given an input sequence $X = \{x_1, x_2, \ldots, x_T\}$, the encoder maps the sequence into a latent representation:

$$h_t = \mathrm{LSTM}(x_t, h_{t-1}) \quad (4)$$

The final hidden state is projected into a latent vector z:

$$z = W_h h_T + b_h \quad (5)$$

The decoder reconstructs the sequence from the latent vector:

$$\hat{x}_t = f_{\text{decoder}}(z) \quad (6)$$

The CNN autoencoder uses one-dimensional convolutional layers to extract local patterns:

$$z = \sigma(W * X + b) \quad (7)$$

$*$denotes convolution and $\sigma$is an activation function. The decoder reconstructs the input using transposed convolution operations:

$$\hat{X} = f_{\text{decoder}}(z) \tag{8}$$

Model Training

Both models are trained to minimize the reconstruction loss using Mean Squared Error (MSE):

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \| X^{(i)} - \hat{X}^{(i)} \|^2 \tag{9}$$

$X^{(i)}$is the original sequence and $\hat{X}^{(i)}$is the reconstructed sequence. Optimization is performed using the Adam optimizer with a learning rate of 0.001.

Anomaly Detection

After training, anomaly scores are computed using reconstruction error:

$$E^{(i)} = \frac{1}{T \cdot F}\sum_{t=1}^{T}\sum_{j=1}^{F}\left(x_{t,j}^{(i)} - \hat{x}_{t,j}^{(i)}\right)^2 \tag{10}$$

A statistical threshold is used to classify anomalies:

$$\tau = \mu_E + 2\sigma_E \tag{11}$$

$\mu_E$and $\sigma_E$represent the mean and standard deviation of reconstruction errors on the training data. A sequence is classified as anomalous if:

$$E^{(i)} > \tau \tag{12}$$

Model Agreement Analysis

To improve reliability, a consensus-based approach is used. A sequence is considered a high-confidence anomaly if both models agree:

$$A_{\text{consensus}} = A_{\text{LSTM}} \cap A_{\text{CNN}} \tag{13}$$

$A_{\text{LSTM}}$and A_{CNN}represent anomaly sets detected by each model.

Algorithm 1: Sequential Anomaly Detection Using LSTM and CNN Autoencoders

Input: dataset D, sequence length T

Output: anomaly labels $A_{\text{LSTM}}, A_{\text{CNN}}, A_{\text{consensus}}$

Process:

Start

Load dataset $D = \{r_1, r_2, \ldots, r_n\}$

Remove irrelevant features

For each transaction r_i:

Extract time features: hour, day, month, weekend

For each account:
  Sort transactions by time
  Compute time difference:
  $\Delta t_i = \frac{t_i - t_{i-1}}{3600}$
Normalize numerical features:
  $x' = \frac{x-\mu}{\sigma}$
Encode categorical features using one-hot encoding
Construct feature matrix X
For each account:
  Apply sliding window with length T
  Create sequence:
  $X^{(i)} = \{x_i, x_{i+1}, \ldots, x_{i+T-1}\}$
Split data into training and validation sets
Train LSTM autoencoder:
  Minimize loss
  $\mathcal{L}_{\text{LSTM}} = \| X - \hat{X} \|^2$
Train CNN autoencoder:
  Minimize loss
  $\mathcal{L}_{\text{CNN}} = \| X - \hat{X} \|^2$
For each sequence $X^{(i)}$:
  Compute reconstruction error:
  $E^{(i)} = \frac{1}{T \cdot F} \sum (x - \hat{x})^2$
Compute thresholds:
  $\tau = \mu + 2\sigma$
If $E^{(i)} > \tau$:
  Label as anomaly
Compute consensus anomaly:
  $A_{\text{consensus}} = A_{\text{LSTM}} \cap A_{\text{CNN}}$
End

Results

Dataset and Sequence Construction

The dataset used in this study consists of 2,512 bank transaction records with 16 original features, including transaction amount, transaction type, location, channel, customer attributes, and account-related information. During preprocessing, irrelevant attributes such as transaction identifiers, device information, and IP addresses were removed to reduce noise and avoid data leakage. Temporal features were extracted from the transaction timestamp, including hour, day of the week, month, and weekend indicators. In addition, a new feature representing the time difference between consecutive transactions for each account was computed to capture behavioral patterns over time. After handling missing values and applying normalization and encoding techniques, the final feature space was expanded to 163 dimensions.

To capture sequential transaction behavior, the dataset was transformed into time-ordered sequences based on account identifiers. Transactions were first

sorted chronologically within each account, and a sliding window approach was applied to construct fixed-length sequences. Each sequence consists of three consecutive transactions, representing short-term behavioral patterns of users. This process resulted in a total of 1,546 sequences, which were then used as input for the deep learning models. The sequential representation enables the models to learn temporal dependencies and detect deviations from normal transaction patterns more effectively.

Model Training Performance

Both LSTM and CNN autoencoder models were trained for 20 epochs using the same training and validation splits to ensure a fair comparison. The learning curves presented in figure 2 show a consistent decrease in both training and validation loss throughout the training process for both models. The LSTM autoencoder achieved a final validation loss of 0.0647, while the CNN autoencoder reached a lower validation loss of 0.0174, indicating a better reconstruction capability on the validation data. In both cases, the gap between training and validation loss remains small across epochs, suggesting stable learning behavior without signs of overfitting. Additionally, the CNN model demonstrates faster convergence, with a more rapid reduction in loss during the early epochs compared to the LSTM model.

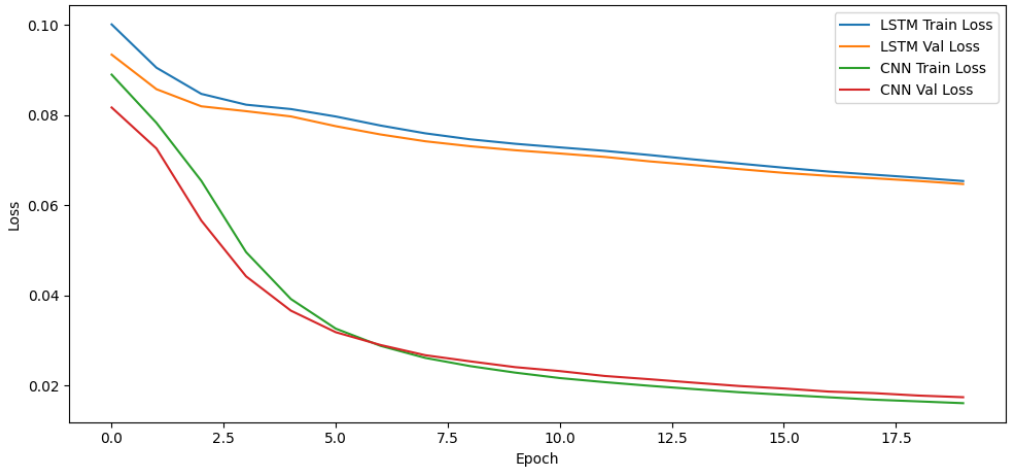


Figure 2 Training and validation loss of LSTM and CNN autoencoders

Reconstruction Error and Anomaly Detection

Reconstruction error is used as the anomaly score. The threshold is defined as:

$$\text{Threshold} = \mu + 2\sigma \tag{14}$$

The distribution of reconstruction errors for both models is presented in figure 3, highlighting the differences in how each model captures normal transaction patterns. The anomaly detection thresholds were determined using the mean and standard deviation of reconstruction errors from the training data, resulting in a threshold of 0.1060 for the LSTM autoencoder and 0.0190 for the CNN autoencoder. The LSTM model exhibits a broader distribution of reconstruction errors, indicating higher variability in how sequences are reconstructed, which reflects its sensitivity to temporal dependencies across transactions. In contrast, the CNN model produces a more concentrated distribution with lower variance,

suggesting more consistent reconstruction of input sequences. This difference implies that the LSTM model may be more responsive to diverse temporal patterns, while the CNN model focuses on capturing localized structures within the sequences.

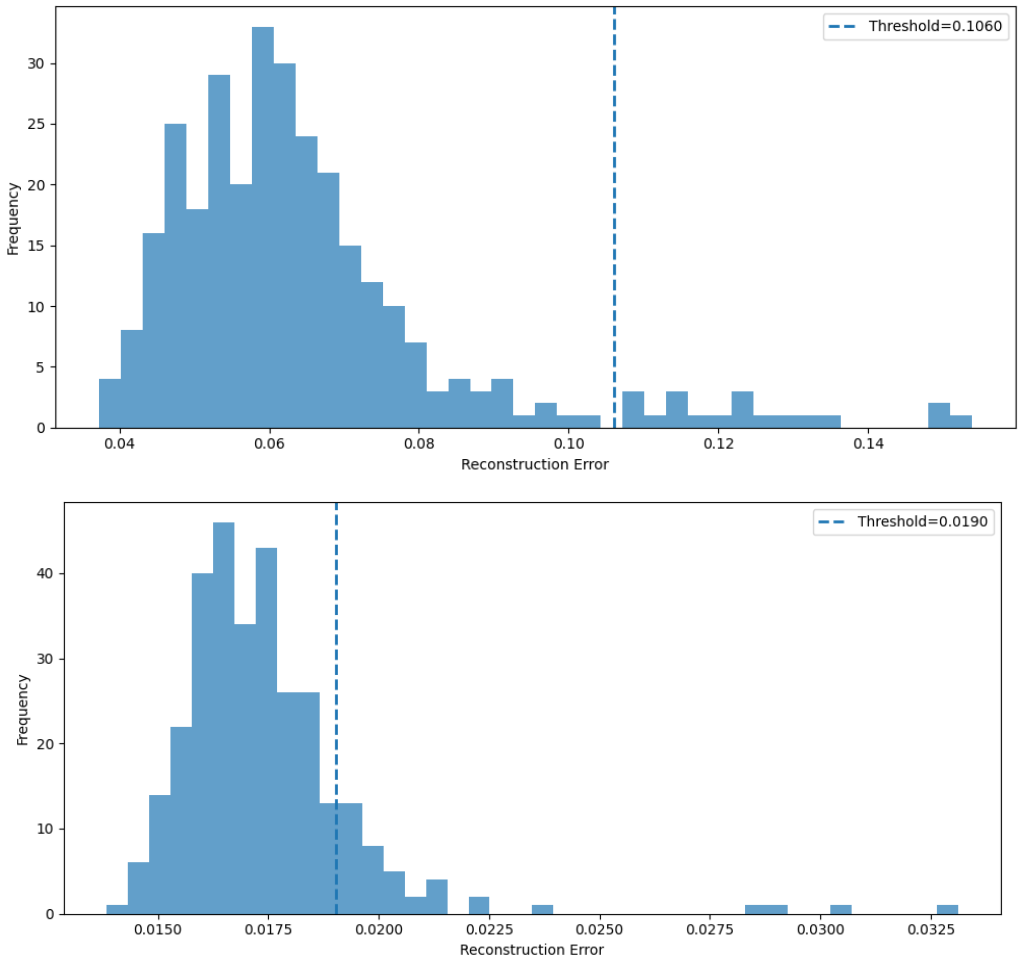


Figure 3 Reconstruction error distribution of the proposed models: (a) LSTM autoencoder, (b) CNN autoencoder.

Anomaly Detection Results

The number of detected anomalies identified by each model is presented in table 1, showing clear differences in detection behavior. The CNN autoencoder detects a total of 42 anomalous sequences, corresponding to 13.55% of the validation data, while the LSTM autoencoder identifies 19 anomalies, representing 6.13% of the data. Among these, 16 sequences are consistently classified as anomalous by both models, indicating a subset of high-confidence anomalies. The higher number of detections by the CNN model suggests that it is more sensitive to deviations in the data, potentially capturing subtle local irregularities within transaction sequences. In contrast, the LSTM model produces fewer detections, indicating a more selective approach that focuses on significant deviations in temporal patterns. The overlap between the two models highlights that, despite their different detection behaviors, there is agreement on a core group of anomalous sequences.

Table 1 Anomaly detection results

Model	Detected Anomalies	Anomaly Rate (%)
LSTM Autoencoder	19	6.13
CNN Autoencoder	42	13.55
Both Models	16	5.16

Model Agreement Analysis

The relationship between reconstruction errors produced by the LSTM and CNN autoencoders is illustrated in figure 4, where each point represents a sequence plotted based on its error values from both models. The scatter plot reveals a dense cluster near the lower-left region, corresponding to normal sequences that are well reconstructed by both models. In addition, several points are distributed outside this cluster, indicating sequences that are identified as anomalous by one or both models. Some points exhibit high reconstruction errors in only one dimension, suggesting that certain anomalies are model-specific and depend on whether temporal dependencies or local patterns are more prominent. A smaller group of points appears in the upper-right region of the plot, where both LSTM and CNN errors are high, representing sequences consistently identified as anomalous by both models. This pattern indicates partial agreement between the models, while also highlighting their complementary detection characteristics.

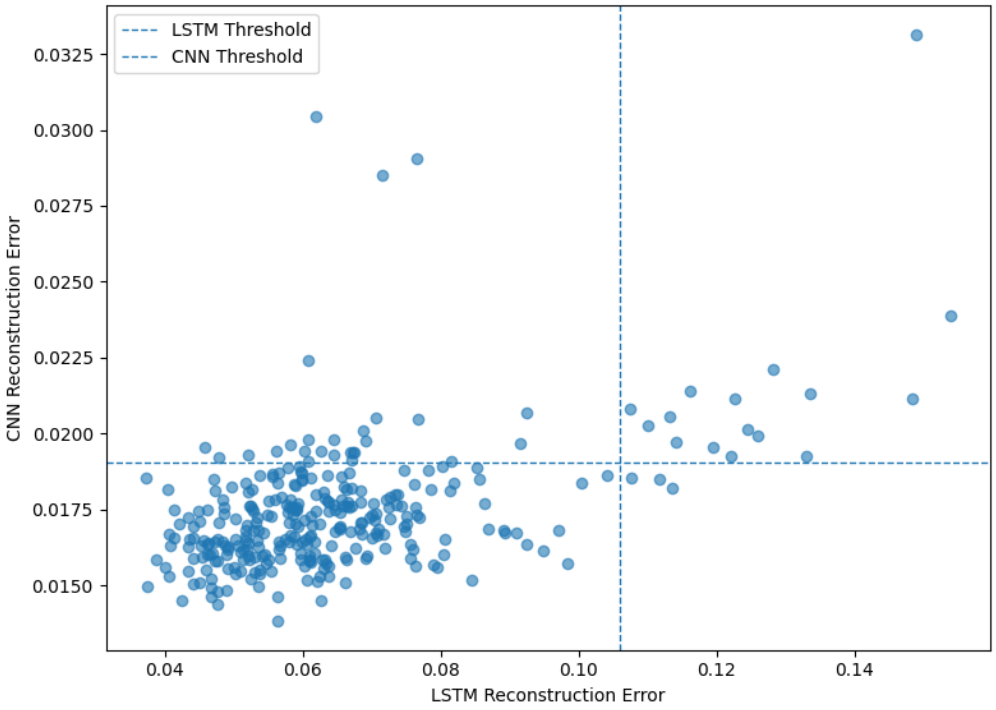


Figure 4 Scatter plot of LSTM vs CNN reconstruction errors

Characteristics of Anomalous Sequences

Statistical differences between normal and anomalous sequences are presented in table 2, highlighting clear distinctions in reconstruction error patterns across both models. Normal sequences exhibit relatively low

reconstruction errors, with mean values of 0.0614 for the LSTM model and 0.0172 for the CNN model, indicating that these sequences are well learned and accurately reconstructed. In contrast, anomalous sequences show substantially higher reconstruction errors, with mean values increasing to 0.1264 for LSTM and 0.0215 for CNN. The average reconstruction error also increases from 0.0393 in normal sequences to 0.0739 in anomalous sequences, demonstrating a consistent deviation from learned patterns. Additionally, the variability of errors is slightly higher in anomalous sequences, reflecting more irregular and unpredictable behavior. These results confirm that anomalous sequences deviate significantly from the normal data distribution, making reconstruction error an effective indicator for anomaly detection in both models.

Table 2 Group statistics for normal vs anomalous sequences

Metric	Normal	Anomaly
Mean LSTM Error	0.0614	0.1264
Mean CNN Error	0.0172	0.0215
Mean Avg Error	0.0393	0.0739

Top Anomalous Sequences

The most anomalous sequences, ranked based on the highest average reconstruction error across both models, are presented in figure 5. These sequences correspond to transaction patterns that deviate most significantly from the learned representations of normal behavior. In many cases, these anomalies are associated with unusually high transaction amounts, irregular temporal gaps between transactions, or uncommon combinations of transaction types and channels. The elevated reconstruction errors indicate that both models struggle to accurately reproduce these sequences, suggesting that such patterns were rarely or never observed during training. Furthermore, sequences identified at the top of the ranking often exhibit consistently high errors in both LSTM and CNN models, reinforcing their classification as strong anomalies. This highlights the effectiveness of using average reconstruction error as a unified metric for prioritizing the most significant deviations in the dataset.

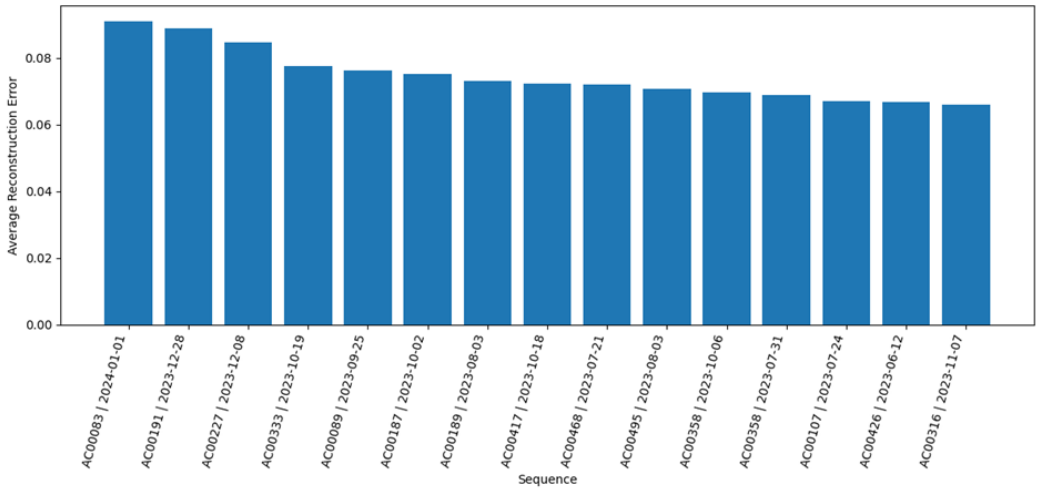


Figure 5 Top anomalous sequences based on reconstruction error

Discussion

This study demonstrates that both LSTM and CNN autoencoders are effective in modeling sequential transaction data and detecting anomalies through reconstruction error, although they exhibit different learning characteristics due to their underlying architectures [21]. The CNN autoencoder achieves lower reconstruction loss and detects a higher number of anomalies, indicating its strong capability to capture local patterns within short transaction sequences [22]. Its relatively compact error distribution suggests consistent reconstruction of normal data, making deviations more distinguishable. However, this behavior also implies that the CNN model may be more sensitive to subtle irregularities, potentially increasing the number of detected anomalies. In contrast, the LSTM autoencoder produces a wider distribution of reconstruction errors and identifies fewer anomalies, reflecting its ability to model temporal dependencies and sequential dynamics [23]. This behavior suggests that LSTM focuses more on significant deviations in temporal structure, such as unusual transaction timing or irregular sequence patterns, resulting in a more selective anomaly detection approach.

The comparison between the two models reveals partial agreement, with a subset of sequences consistently identified as anomalous by both approaches. These shared detections may represent high-confidence anomalies because they are recognized by models that capture different aspects of the data. Meanwhile, model-specific anomalies indicate that each architecture captures distinct characteristics of abnormal behavior. Statistical analysis further supports these observations, showing that anomalous sequences exhibit significantly higher reconstruction errors than normal sequences across both models [24]. In addition, the analysis of the most anomalous sequences suggests that these cases often involve irregular transaction behavior, including large transaction amounts, inconsistent time intervals, or uncommon activity patterns. Despite these promising results, the absence of labeled fraud data limits the quantitative evaluation of detection performance, while threshold selection based on statistical properties may not generalize effectively across different datasets. Future research could address these limitations by incorporating hybrid or ensemble models, exploring longer sequence representations, and integrating advanced architectures to improve detection robustness [25].

Conclusion

This study proposes an unsupervised deep learning framework for anomaly detection in bank transaction data by leveraging sequential modeling through LSTM and CNN autoencoders. By transforming raw transaction records into ordered sequences, the models are able to learn underlying behavioral patterns and identify deviations using reconstruction error without requiring labeled fraud data. The experimental results demonstrate that the CNN autoencoder achieves lower reconstruction loss and detects a higher proportion of anomalies, indicating its effectiveness in capturing local patterns within transaction sequences, while the LSTM autoencoder provides a more selective detection mechanism by modeling temporal dependencies and identifying irregularities in sequence dynamics. The partial agreement between the two models highlights their complementary strengths, where consistently detected anomalies

represent high-confidence deviations from normal behavior. Furthermore, statistical analysis confirms that anomalous sequences exhibit significantly higher reconstruction errors compared to normal sequences, validating the robustness of the proposed approach. Although the absence of labeled data limits quantitative evaluation using traditional metrics, the results provide meaningful insights into transaction irregularities and demonstrate the potential of sequence-based deep learning methods for real-world anomaly detection. Future work may focus on integrating hybrid or ensemble approaches, extending sequence lengths, and exploring more advanced architectures to further enhance detection accuracy and generalization capability.

Declarations

Author Contributions

Conceptualization: M.S.H., R.R.N.F.; Methodology: M.S.H., R.R.N.F.; Software: M.S.H., R.R.N.F.; Validation: M.S.H., R.R.N.F.; Formal Analysis: M.S.H.; Investigation: R.R.N.F.; Resources: M.S.H., R.R.N.F.; Data Curation: R.R.N.F.; Writing – Original Draft Preparation: M.S.H.; Writing – Review and Editing: R.R.N.F.; Visualization: M.S.H.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009, Art. no. 15, doi: 10.1145/1541880.1541882.

[2] M. Goldstein and S. Uchida, “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data,” *PLOS ONE*, vol. 11, no. 4, p. e0152173, Apr. 2016, doi: 10.1371/journal.pone.0152173.

[3] J. West and M. Bhattacharya, “Intelligent financial fraud detection: A comprehensive review,” *Computers & Security*, vol. 57, no. March, pp. 47–66, Mar. 2016, doi: 10.1016/j.cose.2015.09.005.

[4] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *Proc. 2018 Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA, USA, vol. 2018, no. June, pp. 129–134, doi: 10.1109/SIEDS.2018.8374722.

[5] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, Mar. 2021, Art. no. 38, doi: 10.1145/3439950.

[6] M. Markou and S. Singh, "Novelty detection: A review—part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, Dec. 2003, doi: 10.1016/j.sigpro.2003.07.018.

[7] M. Markou and S. Singh, "Novelty detection: A review—part 2: neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, Dec. 2003, doi: 10.1016/j.sigpro.2003.07.019.

[8] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," *Expert Systems with Applications*, vol. 100, no. June, pp. 234–245, Jun. 2018, doi: 10.1016/j.eswa.2018.01.037.

[9] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Proc. 2015 IEEE Symposium Series on Computational Intelligence (SSCI)*, Cape Town, South Africa, vol. 2016, no. June, pp. 159–166, 2015, doi: 10.1109/SSCI.2015.33.

[10] T. Iqbal and S. Qureshi, "Reconstruction probability-based anomaly detection using variational auto-encoders," *International Journal of Computers and Applications*, vol. 45, no. 3, pp. 231–237, 2023, doi: 10.1080/1206212X.2022.2143026.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, vol. 2013, no. December, pp. 1-14, 2013, doi: 10.48550/arXiv.1312.6114.

[12] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA)*, vol. 2014, no. December, pp. 4–11, 2014, doi: 10.1145/2689746.2689747.

[13] I. Lobach and M. Borland, "Long short-term memory networks for anomaly detection in magnet power supplies of particle accelerators," *arXiv preprint arXiv:2405.18321*, vol. 2024, no. May, pp. 1-13, 2024, doi: 10.48550/arXiv.2405.18321.

[14] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, vol. 2016, no. July, pp. 1-5, 2016, doi: 10.48550/arXiv.1607.00148.

[15] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019, doi: 10.48550/arXiv.1901.01588.

[16] D. Kim, S. Park, and J. Choo, "When model meets new normals: Test-time adaptation for unsupervised time-series anomaly detection," in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, vol. 38, no. 12, pp. 13113–13121, Feb. 2024, Art. no.

1462, doi: 10.1609/aaai.v38i12.29210.

[17] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, “Efficient GAN-based anomaly detection,” *arXiv preprint arXiv:1802.06222*, vol. 2018, no. February, pp. 1-13, 2018, doi: 10.48550/arXiv.1802.06222.

[18] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” *arXiv preprint arXiv:1901.03407*, vol. 2019, no. January, pp. 1-50, 2019, doi: 10.48550/arXiv.1901.03407.

[19] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” *arXiv preprint arXiv:1703.05921*, vol. 2017, no. March, pp. 1-12, 2017, doi: 10.48550/arXiv.1703.05921.

[20] S. Karthik, H. V. Supreetha, and S. Sandhya, “Detection of anomalies in time series data,” in *Proc. 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, Bangalore, India, vol. 2022, no. January, pp. 1–5, 2021, doi: 10.1109/CSITSS54238.2021.9683715.

[21] N. Le-Khac, S. Dev, and A. Jurcut, “Network anomaly detection using LSTM-based autoencoder,” *Proc. ACM Symp. QoS Security Wireless Mobile Netw.*, vol. 2020, no. Nov., pp. 1–8, Nov. 2020, doi: 10.1145/3416013.3426457.

[22] S. Ali, A. A. Yazdeen, R. M. Abdullah, and M. H. Younis, “Comparative analysis of CNN for financial fraud detection on tabular transaction data,” *Sci. J. Univ. Zakho*, vol. 14, no. 3, pp. 1–10, Sep. 2026, doi: 10.25271/sjuoz.2026.14.3.1882.

[23] A. Elhalwagy and T. Kalganova, “Hybridization of capsule and LSTM networks for unsupervised anomaly detection on multivariate data,” *Appl. Sci.*, vol. 12, no. 22, pp. 1–15, Nov. 2022, doi: 10.3390/app122211393.

[24] G. Zioviris, K. Kolomvatsos, and G. Stamoulis, “An intelligent sequential fraud detection model based on deep learning,” *J. Supercomput.*, vol. 80, no. 10, pp. 14824–14847, 2024, doi: 10.1007/s11227-024-06030-y.

[25] K. Tang, S. Yao, W. Wu, G.-H. Sun, and Y. Gao, “VAML-Net: Unsupervised anomaly detection for multivariate time series in space-air-ground integrated network (SAGIN) environments through a variational autoencoder and multiresolution LSTM,” *Trans. Emerging Telecommun. Technol.*, vol. 37, no. 1, pp. 1–15, 2026, doi: 10.1002/ett.70342.