Paradigma, Vol. 27, No. 2, September 2025, P-ISSN 1410-5063, E-ISSN: 2579-3500, Page. 65-73
Published: LPPM Universitas Bina Sarana Informatika

PARADIGMA

website: http://jurnal.bsi.ac.id/index.php/paradigma/

Fine-Tuned Autoencoder Neural Network for Anomaly Detection in Accounting Transactions

Nur Alamsyah[1], Hani Fitria Rahmani[2], Wala Erpurini[3], Budiman[4]

[1,4]Fakultas Teknologi Dan Informasi, Universitas Informatika Dan Bisnis Indonesia, Bandung, Indonesia
[2]Sekolah Vokasi, IPB University, Bogor, Indonesia
[3]Fakultas ekonomi dan bisnis, Universitas jenderal achmad yani, Cimahi, Indonesia

ARTICLE INFORMATION

Artikel History:
Received: May 21, 2025
Revised: July 19, 2025
Accepted: Aug. 19, 2025
Available Online: Sept. 15, 2025

Keyword:

Anomaly Detection
Autoencoder Neural Network
Accounting Transactions
Unsupervised Learning
Fraud Prevention

ABSTRACT

Anomaly detection in accounting transactions plays a crucial role in identifying irregularities that may signal fraud, errors, or unusual financial behavior. Traditional rule-based and statistical methods often struggle to detect complex and hidden patterns in large-scale financial datasets. This paper presents a fine-tuned Autoencoder Neural Network for detecting anomalies in structured accounting records. The model processes feature such as date, account type, debit, credit, transaction category, and payment method. Preprocessing includes handling missing values, encoding categorical data, and extracting temporal features. The Autoencoder architecture was optimized using multiple hidden layers and dropout regularization to prevent overfitting. Reconstruction errors were used to determine anomaly scores, with a dynamic threshold set at the 98th percentile. Experimental results show that the model accurately distinguishes normal and anomalous transactions, identifying 2,000 outliers from a total of 100,000 records. Additional analysis indicates that anomalies often occur during weekends or holidays and involve unusual payment methods. These findings demonstrate the potential of the fine-tuned Autoencoder as a scalable and intelligent anomaly detection framework to support auditors and financial analysts in proactive fraud prevention.

Corresponding Author:

Hani Fitria Rahmani,
Sekolah Vokasi,
IPB University,
Jl. Raya Darmaga Kampus IPB, Bogor, Indonesia, 16680,
Email: hanifitria@apps.ipb.ac.id

INTRODUCTION

In recent years, the proliferation of digital financial systems and the adoption of enterprise resource planning (ERP) software have significantly transformed the way organizations record, process, and manage accounting transactions (Jayasuriya & Sims, 2023). These systems generate large volumes of structured transactional data on a daily basis, encompassing elements such as account numbers, debit and credit values, transaction types, payment methods, and timestamps (Dashkevich et al., 2024). While these digital systems have improved operational efficiency and data accessibility, they have also introduced new challenges in ensuring data quality, integrity, and trustworthiness (Yu, 2024).

One of the most pressing concerns is the difficulty in detecting anomalies or irregularities hidden within the vast flow of accounting entries (Mubalaike & Adali, 2018). These anomalies may stem from manual errors, system malfunctions, or even fraudulent activities that are intentionally designed to mimic legitimate transactions (Joshi et al., 2025). In many organizations, audits are still performed periodically and often manually, making them inadequate for catching real-time or subtle discrepancies (Chen et al., 2025). As organizations increasingly rely on automated financial systems, the need for intelligent and scalable anomaly detection mechanisms becomes more critical (Johora et al., 2024).

DOI: https://doi.org/10.31294/p.v27i2.8697

According to Akpan et al. (Akpan, 2024), the detection of financial fraud is one of the most critical and data-intensive applications in the field of data mining, where traditional audit techniques often fall short due to their inability to handle data complexity and scale. (Jacob et al., 2024) further argue that static rule-based fraud detection systems are ineffective in dynamic environments where fraud patterns are constantly evolving.

To overcome these limitations, machine learning (ML) and particularly deep learning (DL) techniques have gained increasing traction in financial anomaly detection (Qatawneh, 2024). One such approach is the use of Autoencoder Neural Networks (AENNs), an unsupervised deep learning model capable of learning compressed representations of data and detecting anomalies based on reconstruction error (Tra et al., 2024). As highlighted by, autoencoders are well-suited for high-dimensional data and perform well in scenarios where labeled data is scarce or unavailable. In the financial context, (Fiore et al., 2019) demonstrate that unsupervised deep learning models outperform traditional supervised models in detecting novel and evolving fraud patterns due to their generalization ability (Huang et al., 2025).

Although autoencoders have been widely used in areas such as network intrusion detection (Kiran et al., 2018), healthcare (Baur et al., 2021), and credit card fraud (Duman and Ozcelik, 2011), the application of fine-tuned autoencoders in structured accounting transaction data remains underexplored (Borgioli et al., 2024). Accounting transactions present unique challenges due to the presence of categorical features (e.g., transaction type, payment method), temporal elements (e.g., date, month, holiday), and dual-natured financial values (debit and credit), which are rarely addressed in existing models (Awosika et al., 2024), (Putrada, Alamsyah, Oktaviani, et al., 2024).

This study proposes a fine-tuned Autoencoder Neural Network for detecting anomalies in accounting transaction datasets. The model is optimized through systematic fine-tuning, including the number of hidden layers, neuron units, activation functions, dropout rates, and training epochs, to better capture domain-specific financial patterns. The anomaly score is derived from the reconstruction error of the autoencoder, and a threshold based on the 98th percentile of the error distribution is used to flag suspicious transactions. The contribution of this research is threefold. First, it applies a deep learning-based unsupervised model—specifically an autoencoder—that is adapted to the structure and characteristics of accounting transaction data, which is an area that remains underrepresented in existing literature. Second, the study introduces a detailed fine-tuning strategy for optimizing the autoencoder's architecture to improve sensitivity and accuracy in anomaly detection. Third, it offers an end-to-end framework that can be implemented without requiring historical fraud labels, making it highly practical for real-world applications where labeled fraud data are often unavailable. This contribution provides a scalable and intelligent solution to support accountants and financial auditors in identifying irregularities proactively and efficiently, thereby contributing to enhanced internal control and financial transparency.

To guide readers in understanding the implementation flow, this study is structured systematically from data acquisition, preprocessing, exploratory analysis, to model training and evaluation. In addition, we include a proposed method diagram illustrating the structure of the Autoencoder neural network used in this study, helping to convey the architecture and process clearly and visually.

RESEARCH METHOD

This section outlines the methodological framework adopted in this study to detect anomalies in accounting transactions using a fine-tuned Autoencoder Neural Network. The entire process is divided into six systematic stages, ranging from data acquisition to model evaluation. Figure 1 illustrates the proposed workflow of the anomaly detection pipeline.

As shown in Figure 1, the methodology begins with Step 1: Dataset Preparation, followed by Step 2: Data Preprocessing, which includes handling missing values, feature extraction, categorical encoding, and normalization. Next, Step 3: Exploratory Data Analysis (EDA) is conducted to examine the distribution of transaction values and detect potential outliers. The processed data then enters Step 4: Anomaly Detection, where an initial autoencoder is trained. In Step 5: Fine-Tuning, the model's architecture is optimized for better reconstruction accuracy. Finally, Step 6: Evaluation measures the performance of the model and interprets the detected anomalies based on reconstruction error thresholds. This structured approach ensures that both the quality of the input data and the learning capability of the model are optimized, resulting in a robust and interpretable anomaly detection system.

1. Data Preparation

The dataset used in this study was obtained from an open-access dataset available on Kaggle, a widely known data science platform that provides real-world and synthetic datasets for machine learning experimentation. The dataset contains detailed financial transaction records structured in tabular format and simulates a wide range of common accounting activities. It consists of 100,000 transactions, where each row represents a single accounting entry involving attributes such as transaction date, account classification, debit and credit values, transaction type, and payment method. These attributes represent key temporal, financial, and categorical aspects, supporting anomaly detection using machine learning. A summary of the attributes along with their descriptions is provided in Table 1.

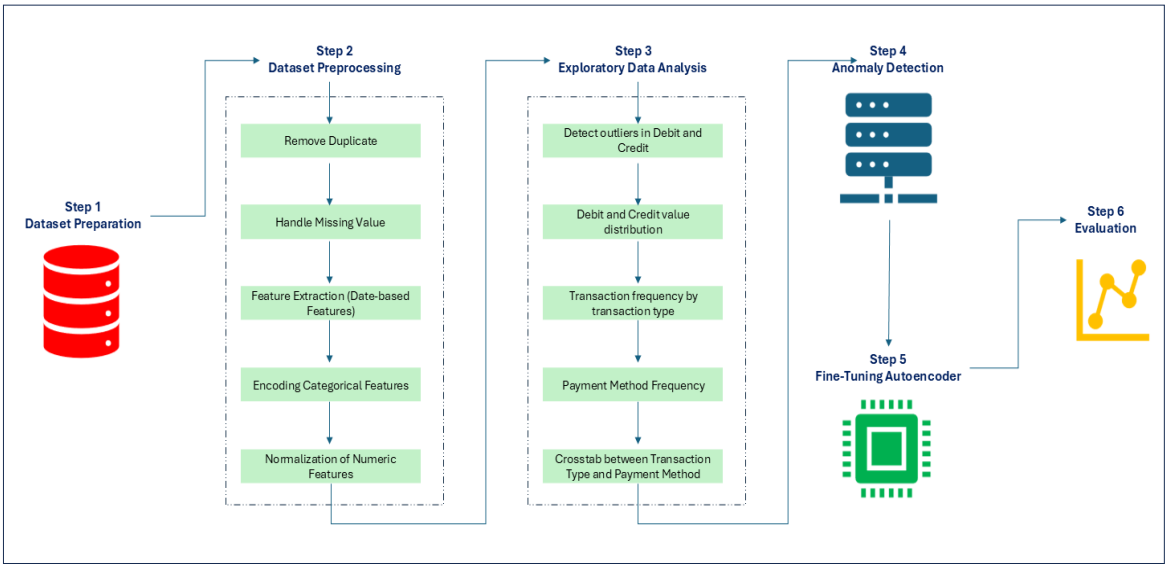

Figure 1. Proposed Method

Figure 1 illustrates the proposed methodological framework, covering all stages from data acquisition to anomaly detection using a fine-tuned autoencoder.

Table 1 presents a detailed description of each attribute included in the financial accounting dataset, highlighting its role in constructing input features for the anomaly detection model.

Table 1. Survey Scale

Attribute	Description
Description	A label assigned to identify individual transactions in the dataset, typically in the format "Transaction X".
Date	The calendar date when the transaction occurred (in YYYY-MM-DD format); also used for deriving time-based features.
Account	The specific financial account associated with the transaction, such as receivables, inventory, or revenue accounts.
Debit	A numeric value representing the amount entered on the debit side of the transaction.
Credit	A numeric value representing the amount entered on the credit side of the transaction.
Category	A higher-level classification of the account, including options such as asset, liability, revenue, or expense.
Transaction_Type	The operational nature of the transaction, categorized into types such as sale, purchase, transfer, or expense.
Customer_Vendor	A masked or anonymized identifier that indicates the customer or vendor involved in the transaction.
Payment_Method	The mode of payment used for the transaction, such as cash, credit card, bank transfer, or check.
Reference	A unique reference number assigned to the transaction, typically used for traceability and record-keeping purposes.

2. Dataset Preprocessing

Prior to modeling, the dataset underwent several preprocessing steps to ensure data quality and compatibility with the anomaly detection framewor (Alamsyah et al., 2024). The preprocessing pipeline consists of the following stages:

a. Data Cleaning:

Duplicate records were removed to avoid redundancy, and missing values were eliminated to maintain the integrity of the dataset (Côté et al., 2024). Additionally, a filtering step was applied to ensure all Debit and Credit values are non-negative, as negative transaction amounts are not valid in typical accounting contexts.

b. Feature Extraction (Date-based):

The Date attribute was converted into a datetime format to enable temporal feature derivation (Cuéllar et al., 2024). From the transaction date, several new features were extracted, including Year, Month, Day, and Weekday. A binary feature Is_Weekend was added to distinguish weekend transactions. Furthermore, transactions occurring on selected public holidays (e.g., 2023-01-01 and 2023-12-25) were flagged using the Is_Holiday attribute.

c. Categorical Feature Encoding:

Categorical variables were encoded to make them suitable for machine learning algorithms (Hikmawati & Alamsyah, 2024). For high-cardinality variables such as Customer_Vendor, Label Encoding was applied. For other categorical attributes including Account, Category, Transaction_Type, and Payment_Method, One-Hot Encoding was used, with the first category dropped to prevent multicollinearity.

d. Numeric Feature Normalization:

To ensure that features such as Debit and Credit contribute proportionally during training, Min-Max Normalization was applied, rescaling the values to a range between 0 and 1.

e. Final Output:

The original Date column was dropped after feature extraction, and the final preprocessed dataset was saved as a CSV file for further analysis and modeling.

These preprocessing steps help transform raw financial transaction data into a structured format, suitable for training a neural network-based anomaly detection model.

3. Exploratory Data Analysis (EDA)

To gain an initial understanding of the dataset and uncover important characteristics and patterns, an Exploratory Data Analysis (EDA) was conducted. This stage aims to provide visual and statistical insights that help in identifying potential anomalies and guiding model development (Putrada, Oktaviani, Fauzan, et al., 2024b). The analysis began by converting the Date column into a datetime format, allowing for the extraction of weekday names to explore temporal transaction trends (Putrada, Alamsyah, Fauzan, et al., 2024). Next, outlier detection and range analysis were performed on Debit and Credit values using boxplots. These plots revealed the presence of potential extreme values, which may represent irregular or suspicious transactions.

Histograms with kernel density estimation (KDE) were generated to examine the distribution of Debit and Credit amounts. The distributions indicate a concentration of transaction values within a certain range, with long tails that may indicate outliers.

In addition, categorical features were explored using count plots. The distribution of transaction types (e.g., Sale, Purchase, Transfer) and payment methods (e.g., Cash, Check, Credit Card, Bank Transfer) was visualized to understand dominant operational behaviors in the dataset.

Finally, a crosstabulation analysis between Transaction_Type and Payment_Method was performed. This matrix-style summary shows the frequency of each payment method used across different transaction types, helping to identify possible patterns or irregularities in transactional behavior.

These insights collectively support the next step in the methodology: anomaly detection using an Autoencoder Neural Network.

4. Anomaly Detection

To detect irregularities within accounting transactions, this study implements an Autoencoder Neural Network, a deep learning approach designed for unsupervised anomaly detection (Putrada et al., 2023). Autoencoders are trained to reconstruct input data with minimal error; thus, when the model encounters unfamiliar or abnormal patterns (i.e., anomalies), the reconstruction error tends to increase significantly (Putrada, Oktaviani, Fauzan, et al., 2024a).

Before model training, the dataset was normalized using Min-Max Scaling to scale numerical features (including Debit and Credit) within the ($[0,1]$) range. Categorical features had already been encoded in the preprocessing stage. Non-numeric identifiers such as Description and Reference were excluded from training. The Autoencoder model architecture consists of:

a. An input layer matching the number of features(n),
b. A bottleneck layer (compressed representation),
c. A reconstruction layer to replicate the input dimensions.
d. Let:
e. ($x \in R^n$) denote the input feature vector,
f. ($\hat{x}$) denote the reconstructed output,
g. ($f(\cdot)$) be the encoder-decoder function learned by the network.

The Mean Squared Error (MSE) between the original input (x) and reconstruction ($\hat{x}$) is used to quantify reconstruction loss:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \qquad (1)$$

This reconstruction error is computed for each transaction. A transaction is classified as anomalous if its reconstruction error exceeds a predefined threshold (τ). The threshold (τ) was determined using the 98th percentile of all MSE values:

$$Anomaly_Score_AE = 1 if MSE > \tau, else 0 \qquad (2)$$

In this study, an anomaly score $Anomaly_Score_AE$ is assigned based on the reconstruction error produced by the Autoencoder model. Specifically, if the MSE between the input and its reconstruction exceeds a predefined threshold τ, the transaction is labeled as anomalous (1); otherwise, it is labeled as normal (0).

The Autoencoder was trained for 20 epochs using the Adam optimizer and MSE loss, with a batch size of 64 and a 10% validation split. After training, a histogram of reconstruction errors was plotted (see Figure 1 to visualize the anomaly threshold and the overall distribution of reconstruction scores.

This threshold-based approach enables the detection of outliers in financial data without requiring prior labeling, making it highly applicable for unsupervised fraud and anomaly screening in accounting systems.

5. Fine Tuning Autoencoder

To improve anomaly detection accuracy, this study implemented a fine-tuned Autoencoder neural network. The model architecture was designed with deeper layers and additional regularization techniques such as Dropout to prevent overfitting (Alamsyah et al., 2024b). The training process used early stopping based on validation loss, ensuring the model did not overfit on the training data.

The dataset was normalized using Min-Max Scaling, and the Autoencoder was trained to reconstruct the original input data. After training, the model computed the reconstruction error for each transaction using Mean Squared Error (MSE). This error reflects how well each transaction aligns with the learned patterns from the normal data.

Transactions with unusually high reconstruction errors above a threshold (τ) were flagged as anomalies. The threshold (τ) was dynamically defined as the 98th percentile of the reconstruction error distribution to capture the top 2% most unusual transactions.

6. Evaluation

The evaluation phase aims to assess the effectiveness of the fine-tuned Autoencoder model in detecting anomalies within accounting transactions. The previously trained Autoencoder model was reloaded alongside the saved anomaly threshold (τ). The dataset was normalized consistently using the same Min-Max Scaler applied during training to ensure accurate reconstruction.

The model then predicted reconstruction outputs and computed the MSE for each transaction. Transactions with MSE exceeding the threshold were labeled as anomalies. The binary outcome was stored in the $\text{Anomaly}_{\text{S}}\text{core}_{(\text{AE})}$ column, where a value of 1 indicates an anomalous transaction and 0 denotes a normal transaction. The evaluation involved three key steps:

1. Anomaly Distribution Analysis:

The total count of anomalies versus normal transactions was calculated and visualized using a histogram of reconstruction errors. The threshold line was also plotted to show the boundary separating normal and anomalous instances.

2. Review of Sample Anomalies:

A subset of transactions flagged as anomalies was examined to provide qualitative insights into suspicious patterns.

3. Transaction Category Insights:

A count plot illustrated the distribution of anomaly labels, offering a visual representation of the proportion of flagged transactions. This step helps identify whether certain periods, transaction types, or payment methods might be more prone to anomalies.

This evaluation not only validates the model's performance but also highlights its practical potential in supporting internal audits and anomaly investigations in financial systems.

RESULTS AND DISCUSSION

Before proceeding with model training and anomaly detection, an exploratory data analysis (EDA) was conducted to better understand the structure, distribution, and potential irregularities within the accounting transaction dataset. This step aims to uncover statistical patterns, detect outliers, and identify characteristics of various transaction types and payment methods. The results of the EDA are summarized in the following visualizations (Figure 2 to Figure 5), which provide valuable insights into the behavior of debit and credit values, as well as categorical distributions. Figure 2 visualizes the distribution and presence of outliers in debit and credit transactions using boxplots.

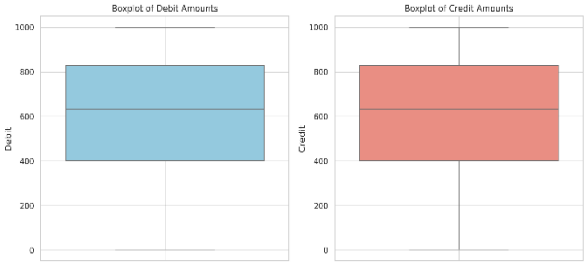

Figure 2. Boxplot of Debit and Credit Amounts

Figure 2 presents boxplots for the Debit and Credit transaction values. These visualizations indicate that most values lie within a normal range, but some extreme values (outliers) are evident—particularly at the higher end. These outliers are of special interest, as they may signal irregularities or anomalous behavior in the financial transactions. Figure 3 shows the distribution of debit and credit values across all transactions, providing insights into transaction frequency and value concentration.

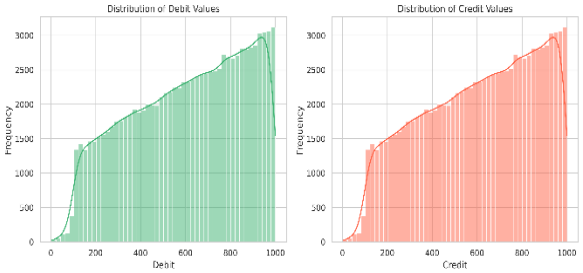

Figure 3. Distribution of Debit and Credit Values

Figure 3 shows the distribution of Debit and Credit values using histograms and KDE (Kernel Density Estimation) curves. Both distributions are right-skewed, suggesting that most transactions occur within small to moderate amounts, while high-value transactions are less frequent but still present. This skewed distribution is typical in financial data and should be considered when building models for anomaly detection. Figure 4 depicts the distribution of transactions categorized by type, identifying the most frequent financial activities in the dataset.

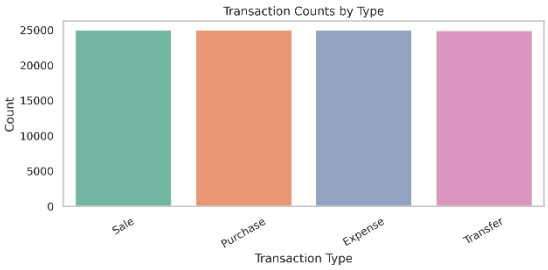

Figure 4. Transaction Counts by Type

Figure 4 illustrates the frequency of transactions categorized by Transaction Type, including Sale, Purchase, Expense, and Transfer. The distribution appears relatively balanced across all categories, which indicates a good representation of various transaction types in the dataset. This balance ensures that the anomaly detection model is not biased toward a particular transaction category. Figure 5 provides a breakdown of transactions based on the payment method used, giving insight into operational preferences.

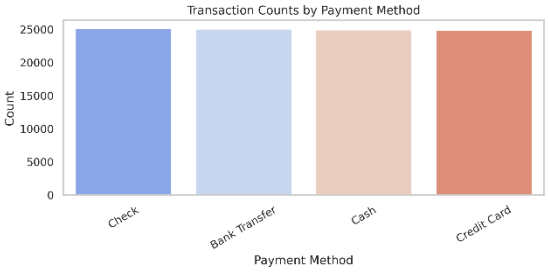

Figure 5. Transaction Counts by Payment Method
Transaction Counts by Type

Figure 5 displays the number of transactions based on the Payment Method, including Check, Bank Transfer, Cash, and Credit Card. Each method is used with relatively equal frequency, reflecting diversity in payment behavior. This variety adds depth to the dataset and can potentially influence anomaly detection outcomes, especially if certain payment methods are more prone to irregularities. Figure 6 displays the reconstruction error produced by the autoencoder model, with a red line indicating the defined anomaly threshold.

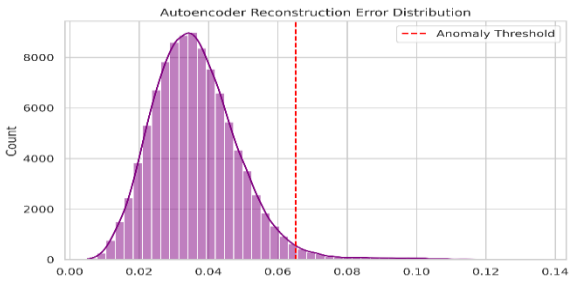

Figure 6. Autoencoder Reconstruction

Figure 6 illustrates the distribution of reconstruction errors generated by the Autoencoder Neural Network. Most transactions exhibit low reconstruction errors, indicating that they follow typical patterns learned by the model. However, a small number of transactions present unusually high reconstruction errors that exceed the anomaly threshold, which is shown as the red dashed line. This threshold is dynamically defined based on the 98th percentile of the reconstruction error distribution.

Transactions with errors greater than this threshold are flagged as anomalies. In this experiment, the Autoencoder successfully identified 2,000 transactions as anomalous from the entire dataset. These anomalies may reflect rare patterns or unusual financial behavior, making them critical for further investigation by financial auditors or fraud analysts.

To enhance the accuracy and generalization ability of the Autoencoder model, a fine-tuning strategy was applied by increasing the number of neurons and incorporating dropout layers to mitigate overfitting. The model was trained with early stopping based on validation loss to ensure optimal performance without excessive training. After the training process, the Autoencoder successfully learned to reconstruct normal accounting transactions with minimal error.

Subsequently, reconstruction errors were calculated for all transactions, and an anomaly threshold was determined using the 98th percentile of the reconstruction error distribution. This threshold value, approximately 0.0547, serves as the decision boundary to classify transactions as normal or anomalous. The trained model and the computed threshold were stored for future evaluation and deployment. This process ensures that the Autoencoder can be consistently reused for real-time or batch anomaly detection tasks in accounting transaction systems.

To evaluate the performance of the fine-tuned Autoencoder model, the saved model and threshold were reloaded and applied to the entire dataset. The model calculated the reconstruction error for each transaction and used the predefined threshold of 0.0547 to classify transactions as either normal or anomalous.

The result of this evaluation indicated that out of the total transactions, 2,000 entries were flagged as anomalies, while the remaining 98,000 transactions were classified as normal. This outcome confirms that the model effectively distinguishes irregularities without over-identifying anomalies.

Figure 7 illustrates the distribution of anomaly labels, clearly showing the significant disparity between normal and anomalous records. This indicates a healthy detection pattern in typical financial datasets, where true anomalies are rare but critical.

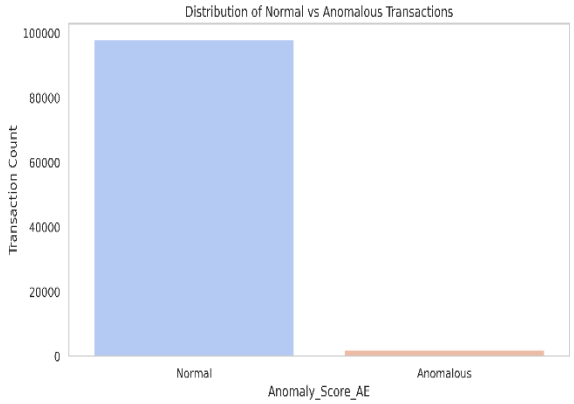

Figure 7. Distribution of Normal vs Anomalous Transactions

This bar chart shows the total number of transactions classified as normal (label 0) and anomalous (label 1). The overwhelming number of normal transactions reflects typical accounting behavior, while a small portion represents transactions with unusual patterns flagged by the Autoencoder.

In addition, Table 2 presents a selection of sample anomalous transactions detected by the model. These samples include key attributes such as debit and credit values, transaction dates, customer/vendor identifiers, and calendar features that may have contributed to their classification as anomalous. Table 2 lists examples of transactions identified as anomalous based on high reconstruction error, offering representative samples of flagged activities.

Table 2. Sample Anomalous Transactions

Index	Debit	Credit	Customer_Vendor	Year	Month	Day	Weekday
101	0.207698	0.207698	89	2023	1	1	6
195	0.825100	0.825100	33	2023	12	25	0
219	0.125840	0.125840	47	2023	2	1	2
227	0.781149	0.781149	40	2023	12	25	0
318	0.242523	0.242523	18	2023	1	1	6
370	0.295680	0.295680	0	2023	11	19	6
431	0.124946	0.124946	6	2023	9	28	3
442	0.363293	0.363293	97	2023	12	25	0
485	0.945799	0.945799	10	2023	1	15	6
511	0.203050	0.203050	91	2023	12	1	4

This study shows that a fine-tuned Autoencoder Neural Network can effectively detect anomalies in accounting transactions. By learning normal transaction patterns, the model identifies irregularities based on reconstruction errors. Using a threshold at the 98th percentile helps minimize false positives while capturing significant deviations.

The model successfully flagged unusual patterns related to debit/credit values, rare payment methods, and transactions on weekends or public holidays. This supports previous findings that Autoencoders are suitable for anomaly detection in structured financial data.

The addition of dropout and early stopping during fine-tuning improved model generalization and prevented overfitting. However, it's important to note that anomalies detected do not always indicate fraud; further analysis or human verification is required.

Overall, the approach demonstrates how deep learning can enhance auditing and financial monitoring processes, offering a data-driven method to support decision-making and risk detection in accounting systems.

CONCLUSION

This study highlights the effectiveness of a fine-tuned autoencoder neural network in detecting anomalies within accounting transaction datasets. Beyond theoretical value, the proposed model holds significant practical relevance in real-world scenarios. For instance, it can assist accountants and auditors in identifying suspicious transactions more efficiently, enhancing the early detection of potential fraud. Moreover, by automating anomaly detection, organizations can improve the integrity of financial reporting processes, reduce operational risks, and optimize auditing workflows. These contributions offer promising avenues for integrating intelligent systems into accounting and financial oversight environments.

REFERENCES

Akpan, D. M. (2024). Artificial Intelligence and Machine Learning. In *Future-Proof Accounting: Data and Technology Strategies* (pp. 49–64). Emerald Publishing Limited. https://doi.org/10.1108/978-1-83797-819-920241007

Alamsyah, N., Kurniati, A. P., & others. (2024a). Airfare Fluctuation Analysis with Event and Sentiment Features by Stacking Ensemble Model. *2024 Ninth International Conference on Informatics and Computing (ICIC)*, 1–6. 10.1109/ICIC64337.2024.10957538

Alamsyah, N., Kurniati, A. P., & others. (2024b). Event Detection Optimization Through Stacking Ensemble and BERT Fine-tuning

For Dynamic Pricing of Airline Tickets. *IEEE Access*. 10.1109/ACCESS.2024.3466270

Awosika, T., Shukla, R. M., & Pranggono, B. (2024). Transparency and privacy: The role of explainable ai and federated learning in financial fraud detection. *IEEE Access*. 10.1109/ACCESS.2024.3394528

Borgioli, N., Aromolo, F., Phan, L. T. X., & Buttazzo, G. (2024). A convolutional autoencoder architecture for robust network intrusion detection in embedded systems. *Journal of Systems Architecture*, *156*, 103283. https://doi.org/10.1016/j.sysarc.2024.103283

Chen, M.-C., Yen, S.-Y., Lin, Y.-F., Tsai, M.-Y., & Chuang, T.-H. (2025). Intelligent Casting Quality Inspection Method Integrating Anomaly Detection and Semantic Segmentation. *Machines*, *13*(4), 317. https://doi.org/10.3390/machines13040317

Côté, P.-O., Nikanjam, A., Ahmed, N., Humeniuk, D., & Khomh, F. (2024). Data cleaning and machine learning: A systematic literature review. *Automated Software Engineering*, *31*(2), 54. https://doi.org/10.1007/s10515-024-00453-w

Cuéllar, S., Santos, M., Alonso, F., Fabregas, E., & Farias, G. (2024). Explainable anomaly detection in spacecraft telemetry. *Engineering Applications of Artificial Intelligence*, *133*, 108083. https://doi.org/10.1016/j.engappai.2024.108083

Dashkevich, N., Counsell, S., & Destefanis, G. (2024). Blockchain financial statements: Innovating financial reporting, accounting, and liquidity management. *Future Internet*, *16*(7), 244. https://doi.org/10.3390/fi16070244

Hikmawati, E., & Alamsyah, N. (2024). Supervised Learning for Emotional Prediction and Feature Importance Analysis Using SHAP on Social Media User Data. *Ingénierie Des Systèmes d'Information*, *29*(6).https://doi.org/10.18280/isi.290622

Huang, H., Wang, P., Pei, J., Wang, J., Alexanian, S., & Niyato, D. (2025). Deep learning advancements in anomaly detection: A comprehensive survey. *IEEE Internet of Things Journal*. DOI : 10.1109/JIOT.2025.3585884

Jacob, L., Thomas, K., & Savithri, M. (2024). AI in Forensics: A Data Analytics Perspective. In *Artificial Intelligence for Cyber Defense and Smart Policing* (pp. 41–60). Chapman and Hall/CRC. https://doi.org/10.1201/9781003251781

Jayasuriya, D. D., & Sims, A. (2023). From the abacus to enterprise resource planning: Is blockchain the next big accounting tool? *Accounting, Auditing & Accountability Journal*, *36*(1), 24–62. https://doi.org/10.1108/AAAJ-08-2020-4718

Johora, F. T., Hasan, R., Farabi, S. F., Alam, M. Z., Sarkar, M. I., & Al Mahmud, M. A. (2024). AI Advances: Enhancing Banking Security with Fraud Detection. *2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP)*, 289–294. DOI: 10.1109/TIACOMP64125.2024.00055

Joshi, R., Pandey, K., & Kumari, S. (2025). Generative AI: A Transformative Tool for Mitigating Risks for Financial Frauds. *Generative Artificial Intelligence in Finance: Large Language Models, Interfaces, and Industry Use Cases to Transform Accounting and Finance Processes*, 125–147. https://doi.org/10.1002/9781394271078.ch7

Mubalaike, A. M., & Adali, E. (2018). Deep learning approach for intelligent financial fraud detection system. *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, 598–603. DOI: 10.1109/UBMK.2018.8566574

Putrada, A. G., Alamsyah, N., Fauzan, M. N., & Oktaviani, I. D. (2024). Pearson Correlation for Efficient Network Anomaly Detection with Quantization on the UNSW-NB15 Dataset. *2024 International Conference on ICT for Smart Society (ICISS)*, 1–6. DOI: 10.1109/ICISS62896.2024.10751550

Putrada, A. G., Alamsyah, N., Oktaviani, I. D., & Fauzan, M. N. (2024). LSTM For Web Visit Forecasting with Genetic Algorithm and Predictive Bandwidth Allocation. *2024 International Conference on Information Technology Research and Innovation (ICITRI)*, 53–58. DOI: 10.1109/ICITRI62858.2024.10698840

Putrada, A. G., Alamsyah, N., Pane, S. F., Fauzan, M. N., & Perdana, D. (2023). Predictive maintenance application on machine overstrain failure with node-red and isolation forest anomaly detection. *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 64–69. DOI: 10.1109/COMNETSAT59769.2023.10420613

Putrada, A. G., Oktaviani, I. D., Fauzan, M. N., & Alamsyah, N. (2024a). CNN Pruning for Edge Computing-Based Corn Disease Detection with a Novel NG-Mean Accuracy Loss Optimization. *Telematika*, *17*(2), 68–83. http://dx.doi.org/10.35671/telematika.v17i2.2899

Putrada, A. G., Oktaviani, I. D., Fauzan, M. N., & Alamsyah, N. (2024b). CNN-LSTM for MFCC-based Speech Recognition on Smart Mirrors for Edge Computing Command.

Journal of Dinda: Data Science, Information Technology, and Data Analytics, *4*(2), 63–74. https://doi.org/10.20895/dinda.v4i2.1504

Qatawneh, A. M. (2024). The role of artificial intelligence in auditing and fraud detection in accounting information systems: Moderating role of natural language processing. *International Journal of Organizational Analysis*. https://doi.org/10.1108/IJOA-03-2024-4389

Tra, V., Amayri, M., & Bouguila, N. (2024). Latent Code Description for Unsupervised AHU Fault Detection Using Adaptive Adversarial Autoencoder. *IEEE Transactions on Automation Science and Engineering*. DOI: 10.1109/TASE.2024.3481211

Yu, G. (2024). Enhancing Accounting Informatization Through Cloud Data Integrity Verification: A Bilinear Pairing Approach. *Journal of the Knowledge Economy*, 1–18. https://doi.org/10.1007/s13132-024-01994-x