GPH-International Journal of Computer Science and Engineering
(Online ISSN 3050-9661)
Volume 9 | Issue 02 | July-December 2026
Published by: Global Publication House
DOI: https://doi.org/10.5281/zenodo.24393743
Journal homepage: https://gphjournal.org/index.php/cse/index

A Model for Anomaly Detection in Financial Institutions Using Deep Learning

UNUKPO, Johnson Urhuemu
Department of Computer Science, Faculty of Science, Rivers State University,
Port Harcourt, Nigeria.

Abstract

The growing sophistication of cyber threats such as data breaches, insider attacks, and fraudulent financial transactions necessitates advanced anomaly detection systems that go beyond the limitations of traditional rule-based and statistical approaches. This research presents the development and evaluation of a deep learning-based anomaly detection model tailored to financial institutions, with a focus on addressing three critical challenges: inadequacy of traditional methods in detecting novel attacks, the impact of severe data imbalance, and the computational demands of deep models in real-time environments. An LSTM autoencoder was implemented within an agile Dynamic Systems Development Method (DSDM) framework, enabling iterative design, stakeholder engagement, and adaptive refinement. The model was trained and tested on financial transaction data, with performance assessed using accuracy, precision, recall, F1-score, and error-based metrics (MAE, RMSE, R^2). Results indicate an overall accuracy of 94%, but only moderate anomaly detection capability, with precision, recall, and F1-score all at 0.40. Daily error metrics remained stable (MAE ≈ 6.2, RMSE ≈ 13.0), though explanatory power was limited ($R^2 \approx 0.03$). Compared to traditional threshold-based systems, the model demonstrated superior adaptability to temporal transaction patterns and provided richer interpretability through visualization of reconstruction errors and anomaly trends. However, the relatively high false-negative rate underscores the need for hybrid approaches integrating supervised techniques and domain knowledge. This study contributes to bridging knowledge gaps in imbalance handling, explainability, and real-time deployment, offering a foundation for more scalable and privacy-preserving anomaly detection frameworks in financial cybersecurity.

Keywords:

Anomaly detection; deep learning; LSTM autoencoder; financial transactions; fraud detection; cybersecurity.

1. Introduction

1.1 Background of the Study

In the evolving landscape of cybersecurity, the role of anomaly detection has gained critical importance as malicious cyber activities escalate in sophistication and frequency. Cybersecurity threats, such as data breaches, insider attacks, and ransomware, continue to challenge organizations globally. These threats demand innovative approaches to identify and mitigate potential security issues before they cause significant harm. Deep learning-based anomaly

© 2026 Author(s). Published by Global Publication House (GPH).
This is an Open Access article distributed under the terms of the CC BY 4.0 License.

detection has emerged as a pivotal research area, blending advanced machine learning algorithms with real-time data processing to detect anomalies. By leveraging vast and complex datasets, deep learning methods can capture nuanced patterns indicative of abnormal behavior that traditional rule-based approaches often fail to detect.

Anomaly detection refers to the identification of rare events or patterns in data that deviate significantly from the norm. In cybersecurity, these anomalies often signify unauthorized access, unusual network traffic, or unexpected system behaviors. Deep learning uses multi-layered neural networks capable of automatically extracting and learning complex features from large datasets. This capability makes it useful for identifying subtle deviations in real-time applications. In financial systems, anomaly detection is particularly important because financial transactions are high-volume and may contain unusual spending patterns, unauthorized transactions, and potential fraudulent behavior.

Traditional approaches to anomaly detection, such as rule-based systems and statistical models, rely on predefined signatures or thresholds. These methods can be effective for known threats but are limited when attack patterns evolve. Data scarcity, severe class imbalance, computational requirements, and privacy concerns also complicate deep learning deployment. The research therefore focuses on developing a scalable and adaptive anomaly detection model for financial transaction monitoring.

1.2 Statement of the Problem

Existing anomaly detection approaches face several challenges. First, traditional rule-based Intrusion Detection Systems and statistical anomaly detection tools rely on predefined thresholds or signatures and may fail to recognize novel or sophisticated attacks. Second, deep learning models, particularly LSTM architectures for time-series data, can demand substantial computational resources and large datasets, creating difficulties for real-time deployment in resource-constrained environments. Third, cybersecurity datasets are typically imbalanced, with benign activities far outnumbering malicious instances, which can result in unreliable detection and high false-positive rates.

Addressing these shortcomings is therefore important for developing a robust, scalable, and efficient anomaly detection framework using deep learning.

1.3 Aim and Objectives

This research aims to develop a model for anomaly detection in financial institutions using deep learning. The objectives are to:

• Design a model that improves upon traditional methods by detecting novel and sophisticated cyber-attacks that traditional rule-based and statistical approaches often miss.

• Address the data imbalance challenge in cybersecurity by implementing techniques that reduce high false-positive rates and enhance system reliability and accuracy.

• Optimise the computational efficiency of the deep learning model, making it viable for real-time anomaly detection in environments with limited storage and processing resources.

1.4 Scope and Significance

The study focuses on developing and evaluating a deep learning-based model for anomaly detection within financial transaction monitoring. It targets irregular or suspicious activities

such as unauthorized transactions, unusual spending patterns, and potential fraudulent behaviors in transaction data. The study involves designing, training, and testing deep learning models, with particular attention to convolutional neural networks and long short-term memory networks to recognize complex patterns in financial datasets.

The project is intended to enhance cybersecurity within the financial sector by monitoring and securing financial transactions against fraudulent activities. Its outcomes are intended to support financial institutions and online transaction platforms where real-time fraud detection is essential, while also contributing to machine learning and cybersecurity research.

2. Related Literature

2.1 Anomaly Detection in Cybersecurity and Financial Transactions

Anomaly detection identifies irregular patterns or behaviors that deviate from expected norms and may signal security threats, system malfunctions, or unknown attacks. In financial systems, transaction monitoring is important for fraud prevention and protection of sensitive assets. The increasing volume and complexity of digital transactions require systems capable of processing transaction streams efficiently and identifying unusual activities such as transactions outside geographical norms or patterns indicative of account takeover.

2.2 Traditional Detection Methods

Rule-based systems operate using predefined thresholds, such as flagging withdrawals exceeding a set limit. Although computationally efficient, these systems are inflexible against adaptive threats. Statistical methods such as Gaussian Mixture Models and Principal Component Analysis identify deviations from probabilistic norms, while machine learning models such as Support Vector Machines and k-Nearest Neighbors learn patterns from historical data. These approaches can struggle with high-dimensional and imbalanced financial datasets.

2.3 Deep Learning, LSTM and Autoencoders

Deep learning can automatically learn features from complex datasets. Convolutional Neural Networks can identify local patterns, while Recurrent Neural Networks and Long Short-Term Memory networks are suited to sequential data and temporal dependencies. LSTMs address the vanishing-gradient problem and can capture long-term dependencies in transaction histories. Autoencoders are unsupervised models that reconstruct input data, with anomalies identified through high reconstruction errors.

2.4 Data Imbalance and Real-Time Processing

Financial datasets are highly imbalanced because fraudulent transactions constitute a small fraction of total transactions. Oversampling, undersampling, and hybrid techniques have been proposed, although oversampling can introduce noise or overfitting and undersampling can remove information. Real-time detection is also challenging because financial systems process high volumes of transactions and deep learning models can require substantial computational resources.

2.5 Explainability, Privacy and Hybrid Approaches

The black-box nature of deep learning can limit adoption in critical financial applications. Explainability is important for trust, regulatory compliance, bias mitigation, and actionable insights. Techniques discussed in the study include feature-importance analysis, activation

visualization, rule extraction, and counterfactual explanations. Privacy-preserving approaches include federated learning, differential privacy, and homomorphic encryption. The literature also discusses hybrid systems that combine traditional rule-based methods with deep learning, allowing predefined anomalies and complex nonlinear patterns to be addressed together.

2.6 Knowledge Gap

The research identifies gaps relating to the limitations of traditional detection methods, data imbalance, computational demands, interpretability, and real-time deployment. The proposed work addresses these issues through an LSTM autoencoder framework, reconstruction-error thresholding, modular implementation, visualization, and a feedback-oriented system design.

3. Materials and Methods

3.1 Research Methodology

The study adopted an agile development framework using the Dynamic Systems Development Method (DSDM) to iteratively design, implement, and refine the deep learning-based anomaly detection system. DSDM emphasizes rapid delivery, continuous stakeholder involvement, incremental development, flexibility, adaptability, and risk mitigation. These characteristics were considered appropriate for a cybersecurity environment in which requirements and threats can evolve.

3.2 Analysis of Existing and Proposed Systems

Existing systems combine data sources, preprocessing, traditional rule-based or statistical detection, deep learning-based detection, and an anomaly detection engine. Traditional methods depend on predefined thresholds and signatures, while deep learning methods automatically learn complex patterns. The proposed system integrates an LSTM-based autoencoder with data preprocessing, anomaly thresholding, real-time processing, visualization, and feedback mechanisms. The design also describes transfer learning, federated learning, and edge computing as strategies for adaptation, privacy, and resource efficiency.

3.3 Data Collection and Dataset

The experimental setup used a dataset named financial_fraud_detection_dataset.csv consisting of approximately 5 million financial transaction records. Each record contains 18 attributes, including transaction metadata such as amount, timestamp, location, and derived features including velocity_score, geo_anomaly_score, and spending_deviation_score. Five numerical features were selected for model training and inference based on their relevance to fraud detection: amount, velocity score, spending_deviation_score, geo_anomaly_score, and time_since_last_transaction.

The dataset was cleaned to remove missing or corrupt entries, and timestamps were normalized for consistent temporal analysis. The study also describes public datasets, simulated data generation, automated data ingestion, normalization, feature extraction, and data balancing as components of the broader data-collection design.

3.4 Preprocessing

The selected numerical features were normalized using MinMaxScaler. To capture temporal dependencies, each input sequence was constructed using a sliding window of 10 consecutive transactions per user, resulting in a three-dimensional input tensor with shape (num_sequences, 10, 5).

3.5 LSTM Autoencoder Architecture

The implemented model is an LSTM-based autoencoder trained to reconstruct normal transaction sequences and flag deviations based on reconstruction error. The architecture consists of an encoder with stacked LSTM layers to learn compact latent representations of normal transaction behavior and a symmetric decoder to reconstruct the input sequence. Reconstruction loss was computed using Mean Squared Error. The model was implemented using TensorFlow and Keras and trained using the Adam optimizer.

The system design describes data ingestion, preprocessing, model inference, real-time processing, monitoring, visualization, and feedback. During inference, a transaction sequence is passed through the model and its reconstruction error is computed. An anomaly is flagged when the reconstruction error exceeds a threshold derived from the 99.5th percentile.

3.6 System Implementation

The system was implemented as a modular architecture combining deep learning, backend APIs, and an interactive dashboard. TensorFlow and Keras were used for the LSTM autoencoder; pandas and NumPy were used for time-series manipulation and numerical transformations; MinMaxScaler from scikit-learn was used for normalization; FastAPI was used for the backend API; and Streamlit was used for the dashboard. Matplotlib and Seaborn were used for visualization. The system also used custom data-ingestion and validation utilities, and the components were managed in a version-controlled environment.

The hardware requirements included an Intel Core i5 or higher processor, at least 8 GB RAM, at least 10 GB free disk space, and an optional NVIDIA GPU with CUDA support. The software environment included Windows 10/11, macOS, or Linux, Python 3.10 or higher, TensorFlow, Keras, pandas, NumPy, scikit-learn, Matplotlib, Seaborn, FastAPI, Streamlit, Git, and Virtualenv.

3.7 Experimental Setup

The model training workflow loaded the preprocessed sequences, split them into training and validation sets, and trained the autoencoder. The source specifies Mean Squared Error as the loss function, Adam as the optimizer, batch size of 128, early stopping based on validation loss, and input shape (None, 10, 5). The training output reported a sequence shape of (4,820,437, 10, 5).

Inference and visualization used the FastAPI backend and Streamlit dashboard. The dashboard provided anomaly summaries, reconstruction-error visualizations, threshold information, evaluation metrics, daily trends, and period comparisons. The 99.5th-percentile reconstruction-error threshold was used for anomaly classification.

3.8 Performance Evaluation Metrics

The study evaluated reconstruction quality using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). MAE measures average reconstruction error, RMSE emphasizes larger errors, and R^2 evaluates how closely predicted values approximate actual values. Classification performance was also examined using accuracy, precision, recall, and F1-score through the confusion matrix.

4. Results

4.1 Global Model Evaluation

Metric	Value
Mean Squared Error (MSE)	171.77
Root Mean Squared Error (RMSE)	13.1
Mean Absolute Error (MAE)	6.21
R^2 Score	0.034

The global evaluation produced an MAE of 6.21, RMSE of 13.1, MSE of 171.77, and R^2 of 0.034.

4.2 Daily Evaluation Metrics

Date	MAE	RMSE	R^2 Score
2025-07-18	6.1	12.8	0.03
2025-07-19	6.25	13.3	0.032
2025-07-20	6.35	13.4	0.029
2025-07-21	6.15	13.0	0.031
2025-07-22	6.17	13.1	0.033

The daily metrics remained relatively stable across the reported dates. RMSE ranged from 12.8 to 13.4, while R^2 remained between 0.029 and 0.033.

4.3 Anomaly Distribution

Device Type	Anomaly Count
Mobile	420
Desktop	300
Tablet	180
ATM	150
POS	90

Hour	Anomaly Rate (%)	Total Transactions	Anomalies Detected
0	2.3	200	5
1	2.5	220	6
2	1.9	180	4
3	1.5	170	3
4	2.0	210	5
5	2.4	190	5

Merchant Category	Anomaly Count
Retail	310
Electronics	250
Groceries	200
Online Services	180

Travel	130

Weekday	Total Transactions	Anomalies Detected	Anomaly Rate (%)
Monday	1500	45	30
Tuesday	1600	50	3.1
Wednesday	1400	42	3.0
Thursday	1550	48	3.1
Friday	1650	51	3.09

4.4 Threshold Sensitivity

Quantile	Threshold	Anomalies	Rate (%)
0.99	0.079234	400	8.0
0.995	0.087543	300	6.0
0.998	0.092321	180	3.6
0.999	0.097832	120	2.4

The threshold sensitivity analysis shows that increasing the quantile threshold reduced the number and rate of detected anomalies. The 0.995 quantile corresponds to a threshold of 0.087543 and 300 detected anomalies in the reported summary.

4.5 Confusion Matrix and Classification Metrics

	Predicted Normal	Predicted Anomalous
Actual Normal	920 (True Negatives)	30 (False Positives)
Actual Anomalous	30 (False Negatives)	20 (True Positives)

The reported confusion matrix produced an overall accuracy of 94%. Precision was 0.40, recall was 0.40, and F1-score was 0.40. The model correctly identified 40% of actual anomalies and missed 60% of genuine fraudulent transactions in the reported matrix.

4.6 Training Results

The reported training run used 4,820,437 sequences with a sequence shape of (4,820,437, 10, 5). The reconstruction loss decreased over the ten reported epochs as follows:

Epoch	Loss	Time (s)
1	0.0134	984
2	7.8672e-04	944
3	2.4031e-04	958
4	1.0821e-04	1060
5	7.8589e-05	927
6	6.1958e-05	1178
7	5.1280e-05	1077
8	4.4133e-05	1015
9	3.8877e-05	701
10	3.4958e-05	778

The reported training output shows a progressive reduction in reconstruction loss, ending at 3.4958e-05 in the tenth reported epoch. The model and scaler were then saved.

5. Discussion

The anomaly detection system developed in this research demonstrates the practical viability of using an LSTM autoencoder for identifying fraudulent financial transactions. The results were visualized and analyzed across device type, time, merchant category, transaction amount, reconstruction error, and daily model metrics.

The device analysis showed that mobile and web-based platforms accounted for a relatively high number of detected anomalies in the reported results. Temporal analysis indicated behavioral clustering across hours, while merchant-category analysis showed differing anomaly frequencies across sectors. The reconstruction-error versus transaction-amount analysis suggested a mild positive correlation, indicating that unusually large or small amounts may increase model uncertainty.

Daily evaluation showed consistent performance with RMSE values between 12.8 and 13.4 and R^2 values between 0.029 and 0.033. The actual-versus-predicted feature overlay showed close alignment in most cases while also displaying deviations at peaks or sudden transitions. The confusion matrix showed 20 true positives, 30 false positives, 30 false negatives, and 920 true negatives, resulting in precision and recall of 0.40.

The evaluation therefore indicates moderate anomaly-classification performance. The false-negative rate remains an important issue for financial applications because undetected fraudulent activity can have significant consequences. The study identifies threshold tuning, domain-specific knowledge, and real-time streaming support as areas for further enhancement.

6. Conclusion and Recommendations

6.1 Conclusion

This study developed and evaluated a deep learning-based model for anomaly detection in financial institutions, with emphasis on the limitations of traditional detection systems, data imbalance, and computational requirements. The LSTM autoencoder captured temporal dependencies and deviations in transaction behavior, while reconstruction-error thresholding provided the basis for anomaly classification. The implemented system also incorporated preprocessing, visualization, a backend API, and an interactive dashboard.

The evaluation produced an overall accuracy of 94%, with precision, recall, and F1-score of 0.40. MAE was approximately 6.2, RMSE approximately 13.0, and R^2 approximately 0.03. The results demonstrate the system's ability to identify anomalous transaction patterns while also showing the need to reduce missed anomalies and improve explanatory power.

6.2 Recommendations

• Cybersecurity teams and financial institutions should consider transitioning from traditional rule-based detection frameworks to advanced deep learning-based systems such as the LSTM autoencoder model implemented in this study.

• Future enhancements should explore privacy-preserving techniques, particularly federated learning and differential privacy, to strengthen data confidentiality and support compliance with privacy regulations including GDPR and the Nigeria Data Protection Act (NDPA).

6.3 Contributions to Knowledge

• Development of an optimized LSTM-based anomaly detection model for real-time use in resource-constrained environments.

• A threshold-based reconstruction-error framework with sequence handling for addressing rare anomalies without relying on oversampling or GANs.

• An interactive Streamlit-based evaluation and visualization interface incorporating anomaly-rate analysis, reconstruction-error trends, and threshold-sensitivity analysis.

References

Aggarwal, C. C. (2015). Outlier analysis. Springer.

Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50–58.

Bansal, S., Gupta, G., & Sharma, M. (2018). Payment card industry data security standards compliance using blockchain. International Journal of Advanced Research in Computer Science, 9(1), 77–84.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, V., Ivanov, V., ... & van Overveldt, T. (2019). Towards federated learning at scale: System design. In Proceedings of Machine Learning and Systems 2019.

Buczak, A. L., & Guven, E. (2017). A survey of data mining and machine learning methods for cybersecurity intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106, 249–259.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1–58.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784–3797.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Figueiredo, F., Silva, T., & Gonçalves, M. (2020). Financial transaction fraud detection: Techniques and challenges. Journal of Financial Crime, 27(3), 715–731.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In Proceedings of the 41st annual ACM Symposium on Theory of Computing (pp. 169–178).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. Artificial Intelligence Review, 22(2), 85–126.

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 100, 234–245.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

Lin, T.-H., & Jiang, J.-R. (2021). Credit card fraud detection with autoencoder and probabilistic random forest. Mathematics, 9(21), 2683.

Liu, F., Zhou, H., & Zhao, J. (2020). Real-time anomaly detection in financial systems using deep learning. Journal of Financial Analytics, 9(4), 321–338.

Maleki, S., Maleki, S., & Jennings, N. R. (2021). Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. Applied Soft Computing, 108, 107443.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR), 54(6), 1–35.

Nguyen, T., Luo, Z., & Zhu, W. (2023). Edge computing for real-time anomaly detection. IEEE Transactions on Cloud Computing, 11(1), 123–136.

Nguyen, T. T., Tahir, H., Abdelrazek, M., & Babar, A. (2020). Deep learning methods for credit card fraud detection. arXiv preprint arXiv:2012.03754.

Schreyer, M., Sattarov, T., Borth, D., Dengel, A., & Reimer, B. (2017). Detection of anomalies in large-scale accounting data using deep autoencoder networks. arXiv preprint arXiv:1709.05254.

Tan, X., Zhang, W., & Xu, Y. (2020). Data augmentation techniques for addressing class imbalance in fraud detection. IEEE Transactions on Knowledge and Data Engineering, 32(6), 1225–1237.

Voigt, P., & Von dem Bussche, A. (2017). The EU General Data Protection Regulation (GDPR): A practical guide. Springer.

Zhou, Y., Deng, Z., & Luo, X. (2019). Challenges and opportunities in cybersecurity anomaly detection using deep learning. Cybersecurity Journal, 14(2), 150–172.

Editorial extraction note: This manuscript has been extracted and condensed solely from the submitted MSc dissertation. No external research content, new data, new results, or new references have been added.