ISSN (e): 2250 – 3005 || Volume, 16 || Issue, 6 || June – 2026 ||
International Journal of Computational Engineering Research (IJCER)

Research and Application of Deep Learning Techniques for System Monitoring

PhuongAnh DT, VanTrong Thai
[1]Lecturer of Information Technology Faculty, Hanoi University of Natural Resources & Environment, Hanoi, Vietnam
[2]School Of Mechanical and Automotive Engineering, Hanoi University of Industries, Hanoi, Vietnam
Corresponding Author: PhuongAnh Dao Thi

ABSTRACT
In the context of rapid digital transformation and the increasing complexity of both cyber threats and system operations, traditional monitoring and intrusion detection systems relying on rule-based and signature-based approaches have shown significant limitations in identifying evolving anomalies and dynamic system behaviors. These challenges are particularly evident not only in network environments but also in modern industrial systems undergoing digitalization.
The research focuses on two representative deep learning architectures, namely Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), to analyze system data at both spatial and temporal levels. The study considers diverse data sources, including network traffic, system logs, and operational data from digitally transformed industrial environments. These datasets, encompassing both flow-based and detailed data representations, are preprocessed through data cleaning, normalization, and feature extraction techniques to ensure model robustness and reliability. Experimental evaluations are conducted using standard benchmark datasets, and model performance is assessed using widely adopted metrics such as Accuracy, Precision, Recall, and F1-score.
The results demonstrate that both models achieve high detection performance, with CNN reaching an average accuracy of approximately 93.45%, while LSTM outperforms with an accuracy of up to 95.12% and superior recall, effectively reducing missed detections. The findings highlight the effectiveness of deep learning in capturing complex patterns in network traffic and reducing false alarm rates compared to traditional approaches.

Date of Submission: 26-05-2026 Date of acceptance: 06-06-2026

I. INTRODUCTION

In recent years, the rapid advancement of digital transformation and the proliferation of interconnected systems have significantly reshaped the global technological landscape. The widespread adoption of cloud computing, Internet of Things (IoT), and smart manufacturing technologies has led to the increasing complexity of modern systems, ranging from network infrastructures to digitally integrated industrial production environments. As a result, these systems have become critical assets supporting economic, educational, and industrial development. However, this increased connectivity has also introduced serious security challenges, as cyber-attacks continue to grow in both frequency and sophistication. Advanced attack types such as Distributed Denial of Service (DDoS), Advanced Persistent Threats (APT), and zero-day exploits pose significant risks to system integrity, confidentiality, and availability.

Traditional intrusion detection systems (IDS), which rely primarily on rule-based or signature-based mechanisms, have proven inadequate in addressing modern cyber threats. These systems are limited in their ability to detect novel attack patterns and often suffer from high false alarm rates and poor adaptability to dynamic network environments. As a result, there is a growing demand for intelligent, adaptive, and data-driven approaches capable of detecting previously unseen attacks and reducing operational overhead for network administrators.

In response to these challenges, machine learning (ML) and, more specifically, deep learning (DL) techniques have emerged as promising solutions for enhancing network security. Deep learning models are capable of automatically extracting hidden patterns and complex relationships from large-scale network traffic data, thus enabling more accurate and robust intrusion detection. Among these models, Convolutional Neural Networks (CNN) have demonstrated strong performance in capturing spatial features of network traffic, while

Long Short-Term Memory (LSTM) networks are particularly effective in modeling temporal dependencies and sequential behaviors associated with cyber-attacks.

Despite the promising potential of deep learning, several challenges remain. Existing studies often rely on outdated or limited benchmark datasets that do not fully reflect modern network environments. Moreover, many models require substantial computational resources, which limits their deployment in real-time or resource-constrained systems. Additionally, issues such as high false positive rates, limited interpretability of deep models, and insufficient evaluation in real-world scenarios continue to hinder practical adoption.

Therefore, this study aims to investigate and apply deep learning techniques to the problem of network intrusion detection, with a focus on improving detection accuracy and adaptability. Specifically, this research develops and evaluates deep learning models based on CNN and LSTM architectures using standardized datasets. The study also emphasizes data preprocessing strategies, feature extraction, and performance evaluation using metrics such as Accuracy, Precision, Recall, and F1-score.

The main contributions of this study can be summarized as follows:

(i) providing a comprehensive analysis of deep learning approaches for network intrusion detection;

(ii) designing and implementing CNN and LSTM-based models for detecting cyber-attacks from network traffic data;

(iii) conducting extensive experiments to compare model performance and identify their respective strengths and limitations; and

(iv) offering insights into the practical feasibility of deploying deep learning-based IDS in real-world environments.

The remainder of this paper is organized as follows. Section 2 reviews related work in intrusion detection and deep learning applications. Section 3 describes the proposed methodology and data processing techniques. Section 4 presents experimental results and discussion. Finally, Section 5 concludes the paper and outlines future research directions.

II. LITERATURE REVIEW

The rapid evolution of cyber threats has led to extensive research in the field of intrusion detection systems (IDS), particularly with the integration of machine learning (ML) and deep learning (DL) techniques. This section reviews existing studies from both traditional and modern perspectives, highlighting key approaches, achievements, and remaining challenges.

2.1 Traditional Intrusion Detection Approaches

Early intrusion detection systems were primarily based on signature-based and rule-based methodologies. These systems rely on predefined patterns or attack signatures to identify malicious activities. While effective in detecting known threats, they suffer from significant limitations in identifying novel or zero-day attacks. Furthermore, such systems often generate high false positive rates due to rigid rule definitions and lack adaptability to dynamic network environments.

To overcome these limitations, statistical and anomaly-based approaches were introduced. These methods attempt to establish a baseline of normal network behavior and detect deviations as potential threats. However, their performance is highly dependent on feature engineering and domain expertise, making them less scalable in complex and large-scale network systems.

2.2 Machine Learning-Based Intrusion Detection

With the advancement of computational power and availability of large-scale datasets, machine learning techniques have been widely applied to intrusion detection. Algorithms such as Decision Trees, Support Vector Machines (SVM), Random Forest, and k-Nearest Neighbors (k-NN) have been used to classify network traffic.

These approaches improve detection accuracy compared to traditional methods and enable automated classification of network behaviors. Studies have shown that ML-based models can effectively detect common attacks such as DoS and probing activities. However, they still rely heavily on manual feature extraction and are limited in capturing complex nonlinear relationships in high-dimensional data. Additionally, their performance degrades when dealing with evolving attack patterns or highly imbalanced datasets.

2.3 Deep Learning Approaches for Intrusion Detection

Deep learning has emerged as a powerful tool for addressing the limitations of traditional and ML-based approaches. By leveraging multilayer neural networks, DL models can automatically learn hierarchical representations from raw or preprocessed data, reducing the need for manual feature engineering.

Among DL architectures, Convolutional Neural Networks (CNN) have been widely used to extract spatial features from network traffic data. CNN-based models are particularly effective in detecting local patterns in

flow-based data and identifying anomalies such as sudden spikes in traffic, which are common in DDoS attacks. However, CNN models often struggle with capturing temporal dependencies in sequential data.

To address time-dependent patterns, Recurrent Neural Networks (RNN) and their advanced variant, Long Short-Term Memory (LSTM), have been proposed. LSTM networks are capable of learning long-term dependencies and are highly effective in analyzing sequential network traffic. Several studies report that LSTM-based models achieve superior detection performance, especially for attacks that occur over time, such as brute force attacks and stealth scanning.

In addition to supervised learning approaches, unsupervised models such as Autoencoders have been utilized for anomaly detection. These models learn the normal behavior of network traffic and identify anomalies based on reconstruction errors, making them suitable for detecting unknown or zero-day attacks.

More recent studies have explored advanced architectures such as Transformer-based models (e.g., BERT) to analyze raw packet payload data. These models leverage self-attention mechanisms to capture complex contextual relationships within sequences, enabling more effective detection of sophisticated attacks embedded within payload data.

2.4 Network Data Representation and Feature Extraction

Data representation plays a crucial role in the performance of intrusion detection systems. Two major approaches have been widely studied: flow-based analysis and packet-based analysis.

Flow-based approaches utilize statistical features such as packet count, byte volume, flow duration, and protocol information. These methods are computationally efficient and suitable for large-scale network monitoring. Tools such as CICFlowMeter have been developed to extract flow features and generate standardized datasets for IDS research.

On the other hand, packet-based approaches analyze raw payload data, providing more detailed insights into network behavior. While this approach enables detection of complex attacks such as malware and application-layer threats, it requires significant computational resources and faces challenges when dealing with encrypted traffic.

2.5 Challenges and Research Gaps

Despite significant progress, several challenges remain in the application of deep learning to intrusion detection:

- Outdated datasets: Many studies still rely on legacy datasets (e.g., KDD99, NSL-KDD), which do not reflect modern network traffic characteristics.
- High computational cost: Deep learning models, especially LSTM and Transformer-based models, require substantial computational resources, limiting real-time deployment.
- Zero-day attack detection: Supervised models struggle to detect unseen attacks due to lack of labeled training data.
- False positive rates: High false alarm rates remain a critical issue, leading to alert fatigue in practical systems.
- Lack of interpretability: Deep learning models are often considered “black boxes,” making it difficult to explain detection results.
- Limited real-world validation: Many studies focus on offline experiments rather than real-time deployment scenarios.

2.6 Research Direction

To address these challenges, recent research trends focus on hybrid models combining multiple architectures (e.g., CNN-LSTM), lightweight models for edge deployment, and advanced learning paradigms such as reinforcement learning and self-supervised learning.

Building upon these insights, this study aims to evaluate and compare deep learning models, particularly CNN and LSTM, for network intrusion detection, while addressing key challenges such as detection accuracy, false alarm reduction, and practical feasibility.

III. METHODOLOGY AND EXPERIMENTAL SETUP

3.1 Overall Framework

This study proposes a deep learning-based framework for network intrusion detection, comprising four main stages: (i) data collection and preprocessing, (ii) feature extraction, (iii) model development, and (iv) performance evaluation.

The framework is designed to systematically transform raw network traffic into structured and meaningful representations suitable for deep learning models. By integrating both data-driven feature learning and advanced modeling techniques, the proposed approach aims to enhance the capability of detecting diverse and evolving cyber-attacks.

Furthermore, the evaluation stage employs standard performance metrics to comprehensively assess the effectiveness, robustness, and generalization ability of the proposed models in real-world network environments.

3.2 Dataset Description

To ensure the reliability and reproducibility of the experimental results, this study utilizes publicly available benchmark datasets commonly used in intrusion detection research, such as NSL-KDD and CIC-IDS datasets. These datasets contain both normal and malicious network traffic, including various types of cyber-attacks:

- Denial of Service (DoS/DDoS)
- Probe (scanning and reconnaissance attacks)
- Remote to Local (R2L)
- User to Root (U2R)

Each dataset consists of multiple features representing network traffic characteristics, such as protocol type, connection duration, packet size, number of bytes transmitted, and other statistical indicators.

The dataset is divided into two subsets: 80% for training and 20% for testing, ensuring unbiased evaluation of model performance.

3.3 Data Preprocessing

Data preprocessing plays a crucial role in improving the performance of deep learning models. The following steps are performed:

- Data Cleaning: Removal of missing, null, and duplicate records to ensure data consistency.
- Encoding: Categorical variables (e.g., protocol type, service) are converted into numerical form using techniques such as one-hot encoding.
- Normalization: Feature scaling (e.g., Min-Max scaling or standardization) is applied to bring all features into a uniform range, improving model convergence.
- Feature Selection: Irrelevant and redundant features are removed to reduce dimensionality and improve computational efficiency.

3.4 Feature Representation

Two main types of network data representation are considered:

- Flow-based features: Include statistical attributes such as packet count, total bytes, flow duration, and traffic rate. These features provide a global view of network behavior.
- Packet-based features: Derived from raw payload data, capturing detailed information about individual packets and enabling deep inspection of attack patterns.

In this study, flow-based features are primarily used due to their efficiency and suitability for large-scale datasets.

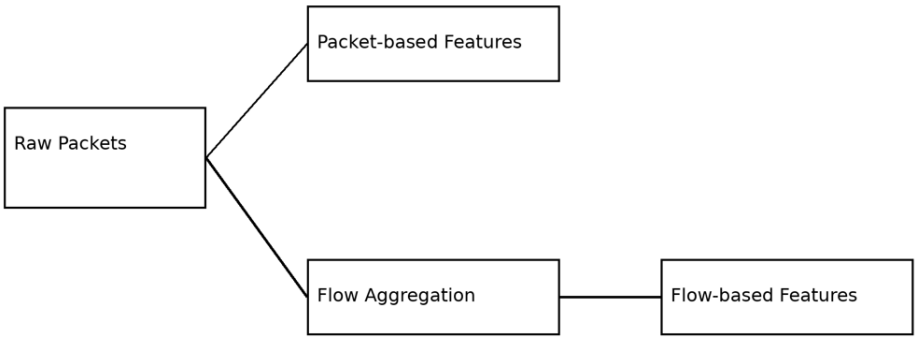

Figure: 3.1 Comparison of packet-based and flow-based feature representation.

3.5 Deep Learning Models

Two deep learning architectures are implemented and compared:

3.5.1 Convolutional Neural Network (CNN)

The CNN model is designed to extract spatial features from network traffic data. The architecture consists of:

- Input layer receiving normalized feature vectors
- Multiple convolutional layers for feature extraction
- Pooling layers to reduce dimensionality
- Fully connected (dense) layers for classification
- Output layer with softmax activation for multi-class classification

CNN is particularly effective in identifying local patterns and correlations within network traffic data.

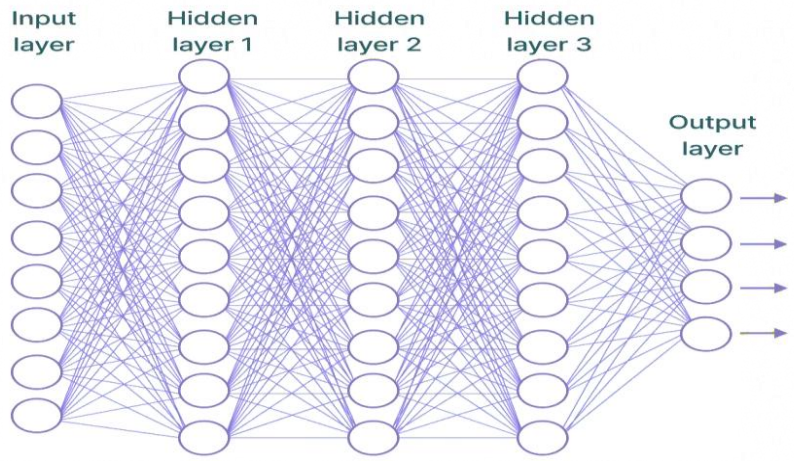

Figure: 3.2 CNN Model

3.5.2 Long Short-Term Memory (LSTM)

LSTM is selected to capture temporal dependencies in sequential network data. The model includes:

- Input layer with sequential data representation
- One or more LSTM layers with memory cells
- Dropout layers to prevent overfitting
- Dense layers for classification

LSTM is especially suitable for detecting time-dependent attack patterns such as brute force or slow scanning attacks.

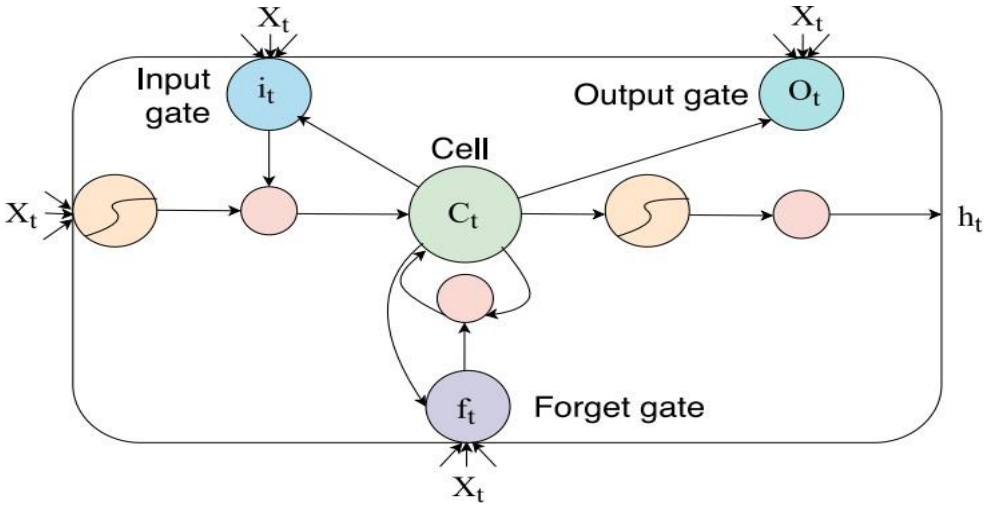

Figure: 3.3 LSTM Model

3.6 Training Process

The training process involves:

- Loss function: Categorical Cross-Entropy
- Optimizer: Adam optimizer
- Batch size: Typically set between 32–128
- Number of epochs: Determined experimentally (e.g., 20–50 epochs)
- Validation split: A portion of training data (e.g., 10–20%) is used for validation

To prevent overfitting, early stopping and dropout regularization techniques are applied.

3.7 Evaluation Metrics

Model performance is evaluated using standard classification metrics:

- Accuracy: Overall correctness of predictions
- Precision: Proportion of correctly predicted attack instances

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

TP (True Positive): The number of positive samples that are correctly predicted.
FP (False Positive): The number of negative samples that are incorrectly predicted as positive.
- Recall: Ability to detect actual attacks (critical in cybersecurity)
Recall $= \frac{TP}{TP+FN}$

- F1-score: Harmonic mean of precision and recall

$$F_1 - Score = 2 \times \frac{\text{Precision } \times \text{ Recall}}{Precision + Recall}$$

The above formula can also be expressed in a more concise form as follows:

$$F_1 - Score = 2 \times \frac{2 \times \text{TP}}{2 \times TP + FP + FN}$$

The F1-score ranges from 0 to 1, where 1 indicates the best performance, representing a model with both high precision and high recall.
- False Positive Rate (FPR): Measures incorrect attack alerts
These metrics provide a comprehensive assessment of detection capability and reliability.
Where:

$$FPR = \frac{FP}{FP + TN}$$

- Positive (P): Instances with values ≥ the threshold.
- Negative (N): Instances with values < the threshold.
- False Positive (FP): The total number of cases where observations belonging to the Negative class are incorrectly predicted as Positive.
- True Negative (TN): The total number of cases correctly predicted as Negative.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Setup

Experiments are conducted in a controlled and reproducible environment to ensure the reliability and fairness of model evaluation. The experimental configuration is summarized as follows:
- Programming language: Python (version 3.9 or later)
- Frameworks and libraries: TensorFlow/Keras for deep learning model development, and Scikit-learn for data preprocessing and evaluation
- Hardware configuration: A GPU-enabled computing system is utilized to accelerate training processes and handle large-scale datasets efficiently
All models are trained and evaluated under consistent settings, including identical data splits, preprocessing procedures, and evaluation metrics. This ensures a fair and objective comparison of model performance.

4.2 Performance Results

The performance of the CNN and LSTM models is summarized as follows:
Table 4.1: Performance Comparison Between Two Experimental Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training time (s)
CNN	93.45	92.80	91.15	91.97	120
LSTM	95.12	94.50	93.80	94.15	350

The LSTM model achieves superior performance, particularly in Recall and F1-score, indicating its effectiveness in detecting sequential attack patterns. CNN also performs well, especially in identifying localized anomalies.

4.3 Discussion

The experimental results indicate that:
- Deep learning models significantly outperform traditional methods in detecting complex network attacks.
- LSTM demonstrates better performance in time-series analysis due to its ability to capture temporal dependencies.
- CNN is effective in detecting spatial patterns but has limitations in modeling sequential behavior.
- Data preprocessing and feature engineering play a critical role in model performance, contributing significantly to detection accuracy.
However, several limitations remain:
- The models require high computational resources, particularly during training.

- Detection performance may vary when applied to real-world traffic data with higher noise and complexity.
- False positives still exist, especially in scenarios where normal traffic resembles attack patterns.

V. CONCLUSION

This study presents a comprehensive investigation into the application of deep learning techniques for network intrusion detection, with a focus on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models. By integrating data preprocessing, feature engineering, and model optimization, the proposed framework demonstrates strong capability in detecting a wide range of cyber-attacks within network traffic data.

The experimental results confirm that both models achieve high performance, with CNN reaching an accuracy of approximately 93.45% and LSTM achieving up to 95.12%. Notably, LSTM shows superior performance in terms of recall and F1-score, indicating its effectiveness in capturing temporal dependencies and identifying sequential attack patterns. In contrast, CNN proves to be efficient in extracting spatial features and detecting localized anomalies in network traffic. These findings highlight that different deep learning architectures offer complementary strengths in intrusion detection tasks.

In addition, the study emphasizes the critical role of data preprocessing and feature selection in enhancing model performance. Experimental observations suggest that well-structured input data significantly improves convergence speed and detection accuracy. The results also demonstrate that deep learning models can reduce false alarm rates while maintaining high detection capability, thereby addressing key limitations of traditional rule-based intrusion detection systems.

Despite these promising outcomes, the study is conducted in a controlled experimental environment using benchmark datasets, which may not fully reflect the complexity and variability of real-world network traffic. Challenges such as encrypted data, highly dynamic attack patterns, and scalability issues remain open for further exploration. Nevertheless, the findings provide strong evidence for the feasibility of deploying deep learning-based intrusion detection systems in modern cybersecurity infrastructures.

REFERENCES

[1]. Ahmad, T. (2020), "CICFlowMeter-V4.0: Network Traffic Flow Generator and Feature Extractor", University of New Brunswick, Canada.
[2]. Ali, W. A., et al. (2021), "A Review of Current Machine Learning Approaches for Anomaly Detection in Network Traffic", IEEE Access, vol. 9, pp. 15201-15222.
[3]. 8. Hassan, M. M. (2024), "Deep Reinforcement Learning for Adaptive Cyber Defense in Smart Grids", Journal of Network and Computer Applications, vol. 210.
[4]. Liu, H., & Lang, B. (2020), "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey", Applied Sciences, 10(5), 1675.
[5]. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2022), "Toward generating a new intrusion detection dataset and intrusion traffic characterization", Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP).
[6]. Vinayakumar, R., et al. (2023), "Deep Learning for Intelligent Healthcare Systems: Network Security Perspectives", Future Generation Computer Systems, vol. 142, pp. 200-215.
[7]. Wang, W., et al. (2021), "Malware Traffic Classification Using Convolutional Neural Networks for Representation Learning", International Conference on Information Networking (ICOIN).
[8]. Zhang, J. (2022), "Bidirectional Encoder Representations from Transformers (BERT) for Network Intrusion Detection", ArXiv preprint arXiv:2203.04567.