Afro-Asian Journal of Scientific Research (AAJSR)
المجلة الأفروآسيوية للبحث العلمي
E-ISSN: 2959-6505
Volume 4, Issue 2, 2026
Page No: 77-83
Website: https://aajsr.com/index.php/aajsr/index
SJIFactor 2024: 5.028 ISI 2025: 0.915 معامل التأثير العربي (AIF) 2025: 0.76

A Lightweight Deep Learning Model for Real-Time Anomaly Detection in Network Traffic

Saad A S Abdurehiem[1*], Anees Almabrouk Salih[2]
[1,2]Department of Information Technology, High Institute of Science and Technology, Emsaad, Libya

إطار تعلم عميق خفيف وعالي الكفاءة لإكتشاف الشذوذ في حركة الشبكات في الزمن الحقيقي

سعد عبد الرحيم سعد[1*]، أنيس المبروك صالح[2]
[1،2]قسم تقنية المعلومات، المعهد العالي للعلوم والتقنية، أمساعد، ليبيا

*Corresponding author: Saad.A.S.Abdulrahiem@histe.edu.ly

Received: February 13, 2026	Accepted: March 28, 2026	Published: April 08, 2026

Abstract:
This paper introduces a computationally efficient deep learning framework for real-time anomaly detection in network traffic environments characterized by high volume and dynamic behavior. Unlike conventional intrusion detection systems that rely on static signatures or resource-intensive architectures, the proposed model integrates a compact Convolutional Neural Network (CNN) with a streamlined Long Short-Term Memory (LSTM) module to jointly capture spatial and temporal characteristics of network flows. The model is specifically optimized for low-latency inference, making it suitable for deployment in resource-constrained environments such as IoT and edge networks. Experimental validation on the UNSW-NB15 and CIC-IDS2017 datasets demonstrates that the proposed approach achieves an accuracy of 97.8% while maintaining a significantly reduced false positive rate and computational overhead. The results indicate that the proposed architecture effectively balances detection performance and efficiency, offering a practical and scalable solution for modern cybersecurity systems requiring real-time responsiveness.

Keywords: Anomaly Detection, Deep Learning, Real-Time, Network Traffic, CNN-LSTM.

الملخص:
يقدم هذا البحث إطارًا قائمًا على التعلم العميق يتميز بخفة الوزن والكفاءة العالية، بهدف اكتشاف الشذوذ في حركة بيانات الشبكات في الزمن الحقيقي، خاصة في البيئات التي تتسم بكثافة البيانات والتغير الديناميكي المستمر. وعلى خلاف أنظمة كشف التسلل التقليدية التي تعتمد على التواقيع الثابتة أو النماذج ذات الاستهلاك المرتفع للموارد، يعتمد الإطار المقترح على دمج شبكة عصبية التفافية (CNN)مدمجة مع نموذج الذاكرة طويلة قصيرة المدى (LSTM)بصورة مبسطة وفعّالة، مما يتيح التقاط الخصائص المكانية والزمانية لتدفقات الشبكة بشكل متكامل. وقد تم تحسين هذا النموذج لتحقيق استجابة سريعة بزمن تأخير منخفض، الأمر الذي يجعله مناسبًا للتطبيق في البيئات محدودة الموارد مثل إنترنت الأشياء (IoT) وشبكات الحافة (Edge Networks). أظهرت نتائج التقييم التجريبي باستخدام مجموعتي البيانات UNSW-NB15 و CIC-IDS2017أن النموذج المقترح يحقق دقة تصل إلى 97.8%، مع انخفاض ملحوظ في معدل الإنذارات الخاطئة وتقليل العبء الحاسوبي. تشير هذه النتائج إلى أن البنية المقترحة تحقق توازنًا فعّالًا بين دقة الاكتشاف والكفاءة التشغيلية، مما يجعلها حلاً عمليًا وقابلًا للتوسع لدعم أنظمة الأمن السيبراني الحديثة التي تتطلب استجابة فورية وفعّالة في الزمن الحقيقي.

الكلمات المفتاحية: اكتشاف الشذوذ؛ التعلم العميق؛ الزمن الحقيقي؛ حركة الشبكات، CNN-LSTM.

Introduction:

With the rapid expansion of network infrastructures and the growing adoption of digital services, exposure to cyber threats such as Denial of Service (DoS), port scanning, Remote-to-Local (R2L) attacks, and malware has significantly increased. Traditional Intrusion Detection Systems (IDS) often rely on static rules or signature-based methods, which are limited in detecting unknown or sophisticated attacks. These systems also face challenges in processing high-volume network traffic, potentially causing delayed responses and security breaches [1,2].

Recently, deep learning-based approaches have gained attention due to their ability to automatically learn complex patterns from network traffic. Convolutional Neural Networks (CNNs) are effective in capturing spatial correlations among features, while Long Short-Term Memory (LSTM) networks are suited for modeling temporal dependencies, which is crucial for detecting evolving attacks. Hybrid architectures, such as CNN-LSTM, leverage the strengths of both networks to improve detection performance while addressing computational constraints in real-time and resource-limited environments [3–10].

This study proposes a lightweight CNN-LSTM model for real-time network anomaly detection, emphasizing high detection accuracy, low false positive rates, and suitability for IoT and edge computing scenarios.

Related Work:

Recent research indicates that deep learning methods significantly outperform traditional IDS approaches. CNNs are commonly used to extract spatial feature relationships in network data, whereas LSTM networks effectively model temporal sequences, which helps in detecting complex attacks [1–5]. Hybrid models combining CNN and LSTM have been proposed to capture both spatial and temporal patterns. These architectures achieve higher accuracy and reduced false positive rates compared to standalone CNN or LSTM networks [4, 5].

In IoT and edge computing environments, lightweight models are developed to maintain real-time processing capabilities while handling high-throughput network traffic [7,9,10]. Despite these advances, designing hybrid architectures that balance detection performance and computational efficiency remains a challenge, motivating the proposed CNN-LSTM framework in this study.

Problem Statement:

Despite notable advances in intrusion detection systems, several critical issues remain unresolved:

- High-throughput network traffic is often challenging for existing IDS to handle in real time, which can degrade performance.
- Many deep learning solutions, while accurate, are computationally heavy, limiting deployment in resource-constrained environments like IoT devices.
- Elevated false positive rates can overwhelm security personnel and reduce operational efficiency.

These challenges indicate the need for a framework that can provide accurate, efficient, and real-time anomaly detection.

Research Objectives:

This study aims to develop and evaluate a lightweight deep learning model for real-time network anomaly detection. **The main objectives are:**

- To design a hybrid CNN-LSTM model optimized for minimal computational load.
- To maximize detection accuracy while reducing false positives.
- To ensure the model can operate under real-time constraints in high-speed networks.
- To benchmark the proposed model against standalone CNN and LSTM architectures.
- To demonstrate the scalability and feasibility of the model for deployment in IoT and edge computing environments.

Methodology:

Architectural Overview:

The proposed model is based on a hybrid deep learning architecture that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to effectively capture both spatial and temporal characteristics of network traffic data.

The architecture is designed with a focus on computational efficiency and real-time applicability. By combining lightweight convolutional layers with a compact recurrent structure, the model achieves a balance between detection accuracy and processing speed.

The overall system transforms raw network traffic into structured feature representations, which are then processed through sequential learning stages to produce a final classification output indicating normal or anomalous behavior.

Data Preprocessing:

To ensure high-quality input for the model**, the raw network traffic data undergoes a series of preprocessing steps:**

- **Data Cleaning:** Removal of incomplete, duplicate, or corrupted records to enhance dataset reliability.
- **Normalization:** Numerical features are scaled using min-max normalization to ensure stable and efficient training.
- **Categorical Encoding:** Non-numeric features are converted into numerical representations using one-hot encoding.
- **Feature Structuring:** The processed data is reshaped into a format suitable for CNN input.

These steps help reduce noise, improve convergence, and enhance overall model performance.

CNN Feature Extraction:

The convolutional component of the model is responsible for extracting spatial features from network traffic data.

The CNN applies convolutional filters to identify local patterns and correlations among input features. Activation functions introduce non-linearity, allowing the model to learn complex feature representations. A pooling layer is used to reduce dimensionality while preserving important information.

Mathematically, the one-dimensional convolution operation can be expressed as:

$$y_i = \sum_{j=0}^{K-1} w_i \cdot x_{i+j} + b$$

where x is the input, w represents the filter weights, K is the kernel size, and b is the bias.

A non-linear activation function such as ReLU is applied:

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{max}(\mathbf{0}, \boldsymbol{x})$$

The CNN architecture is intentionally lightweight, ensuring reduced computational cost while maintaining effective feature extraction capabilit

LSTM Temporal Modeling:

Following feature extraction, the processed data is passed to an LSTM layer to capture temporal dependencies in network traffic sequences.

The LSTM network utilizes memory cells and gating mechanisms to retain relevant information over time. The internal operations of an LSTM cell are defined as follows:

$$\boldsymbol{ft} = \boldsymbol{\sigma}(\boldsymbol{W_f}[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_f}$$

$$\boldsymbol{i_t} = \boldsymbol{\sigma}(\boldsymbol{W_i}[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_i}$$

$$\widetilde{\boldsymbol{C_t}} = \boldsymbol{tanh}(\boldsymbol{W_c}[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_c}$$

$$\boldsymbol{C_t} = \boldsymbol{f_t} \cdot \boldsymbol{C_{t-1}} + \boldsymbol{i_t} \cdot \widetilde{\boldsymbol{C_t}}$$

$$\boldsymbol{h_t} = \boldsymbol{o_t} \cdot \boldsymbol{tanh}(\boldsymbol{C_t})$$

These equations enable the model to capture long-term dependencies and temporal attack patterns effectively.

Input and Output Shape Analysis:

To ensure reproducibility and clarity, **the dimensional transformations across the model are defined as follows:**

- **Input Layer :** (N, F)
- **After Reshape :** $(N, F, 1)$
- **After Conv1d :** $(N, F, 32)$
- **After Maxpooling :** $(N, F, 2{,}32)$
- **After LSTM :** $(N, 64)$
- **After Dense :** $(N, 32)$
- **Output Layer :** $(N, 2)$

This transformation pipeline ensures efficient feature compression and structured learning.

Training Strategy:

The model is trained using the Adam optimization algorithm due to its efficiency and adaptive learning rate capabilities.

Key training configurations include:

- Fixed number of training epochs.
- Mini-batch gradient descent with a defined batch size.
- Learning rate tuning for stable convergence.

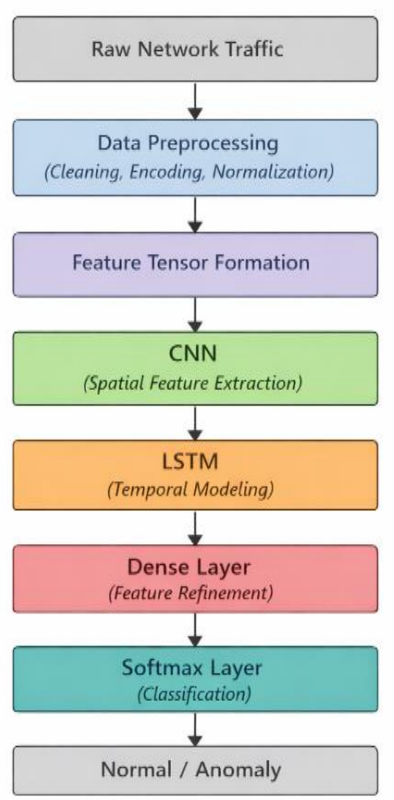
To improve generalization and reduce overfitting, regularization techniques such as dropout may be applied.

Datasets and Experimental Setup:

T**he proposed model is evaluated using two widely recognized benchmark datasets:**

- **UNSW-NB15:** Provides a diverse set of modern attack scenarios and realistic network traffic patterns.
- **CIC-IDS2017:** Includes multiple attack types and detailed traffic features for comprehensive evaluation.

These datasets differ in size, feature dimensions, and attack diversity, enabling a robust assessment of the model's generalization capability.



System Workflow:

The operational workflow of the proposed system follows a sequential pipeline:

Raw Network Traffic → Data Preprocessing → Feature Structuring → CNN Feature Extraction → LSTM Temporal Modeling → Dense Layer → Softmax Classification → Output (Normal / Anomaly)

This pipeline ensures efficient transformation of raw data into meaningful predictions in a real-time environment.

Implementation Environment:

The model is implemented using a modern software stack to ensure efficiency and reproducibility:

- Programming Language: Python 3.10.
- Deep Learning Framework: TensorFlow 2.12 with Keras API.
- Data Processing: Pandas and NumPy.
- Machine Learning Tools: Scikit-learn.
- Visualization: Matplotlib and Seaborn.
- Development Environment: Jupyter Notebook / Google Colab.

The experiments are conducted on a system equipped with Intel i7 CPU, 16GB RAM, and optional NVIDIA GTX 1650 GPU.

Model Architecture:
The detailed architecture of the proposed model is summarized in Table 1.

Table (1): The detailed architecture of the proposed model

Layer Type	Output Shape	Parameters	Description
Input Layer	(None, 42/78)	0	Network traffic features
Reshape Layer	(None, X, 1)	0	Prepare input for CNN
Conv1D	(None, X, 32)	~1,000	Feature extraction
MaxPooling1D	(None, X/2, 32)	0	Dimensionality reduction
LSTM	(None, 64)	~25,000	Temporal learning
Dense	(None, 32)	~2,000	Feature transformation
Dropout	(None, 32)	0	Regularization
Output Layer	(None, 2)	~66	Softmax classification

The total number of parameters is approximately **30,000**, which confirms the lightweight nature of the proposed model.

Complexity Analysis:

The computational complexity of the proposed model consists of two main components:

- **CNN Complexity:** $O(N \cdot F \cdot K)$.
- **LSTM Complexity:** $O(T \cdot H^2)$**.**
- **Overall Complexity:** $O(N \cdot F \cdot K + T \cdot H^2)$**.**

Despite integrating both CNN and LSTM components, the model remains computationally efficient due to its lightweight architecture, making it suitable for real-time deployment.

Results and Discussion:

To evaluate the effectiveness of the proposed model, standard classification metrics are used, including Accuracy, Precision, Recall, and F1-Score.

Table (2): Performance Comparison

Metric	CNN	LSTM	CNN-LSTM (Proposed)
Accuracy	93.8%	95.4%	97.8%
Precision	94.1%	95.7%	98.2%
Recall	92.5%	94.9%	97.1%
F1-Score	93.3%	95.3%	97.6%

Performance Analysis:

The experimental results demonstrate that the proposed CNN-LSTM model significantly outperforms standalone CNN and LSTM models across all evaluation metrics.

- The **accuracy (97.8%)** indicates superior classification capability.
- The **precision (98.2%)** reflects a low false positive rate.
- The **recall (97.1%)** shows effective detection of anomalous traffic.
- The **F1-score (97.6%)** confirms a balanced performance.

These results highlight the advantage of combining spatial and temporal learning in a unified architecture.

Confusion Matrix Analysis:

Table (3): Confusion Matrix

Actual / Predicted	Normal	Anomaly
Normal	98.1%	1.9%
Anomaly	2.7%	97.3%

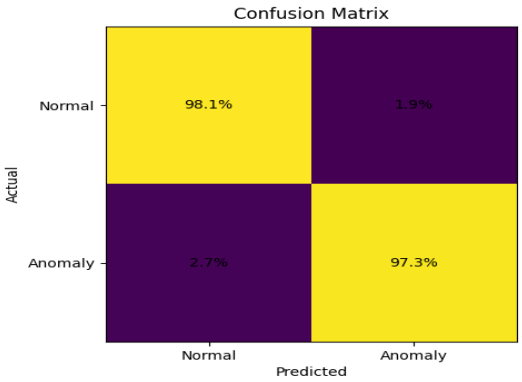


Figure (3): Confusion Matrix

Analysis:

- The model correctly classifies **98.1% of normal traffic**, indicating strong stability.
- The **false positive rate (1.9%)** is very low, which is critical in real-world systems.
- The detection rate for anomalies reaches **97.3%**, showing high sensitivity.
- The **false negative rate (2.7%)** is minimal, reducing the risk of undetected attacks.

This confirms that the model is reliable for real-time deployment.

ROC Curve Analysis:

The Receiver Operating Characteristic (ROC) curve is used to evaluate the classification performance across different thresholds.

The proposed model achieves a high Area Under the Curve (AUC), indicating strong discriminative capability between normal and anomalous traffic.

A high AUC value confirms that the model effectively minimizes both false positives and false negatives.

Training Performance:

To analyze the learning behavior of the model, training and validation accuracy and loss are monitored over epochs.

- The training accuracy increases steadily and stabilizes near optimal values.
- ;8The validation accuracy follows a similar trend, indicating good generalization.
- The loss decreases consistently, confirming stable convergence.

This demonstrates that the model does not suffer from overfitting or underfitting.

Computational Efficiency:

In addition to detection performance, **the proposed model demonstrates strong computational efficiency:**

- Total parameters: approximately **30K only**.
- Lightweight architecture suitable for real-time systems.
- Reduced inference time compared to deep LSTM models.

This makes the model ideal for IoT and edge environments.

Comparison with Existing Methods:

Table (4): Comparison with State-of-the-Art Methods

Study	Model	Dataset	Accuracy	Key Advantage
Wang et al. (2023)	CNN	UNSW-NB15	93.5%	Spatial features
Mennour et al. (2022)	LSTM	CIC-IDS2017	95.1%	Temporal modeling
Ben Said et al. (2023)	CNN-BiLSTM	SDN Dataset	96.8%	Hybrid approach
Rafique et al. (2024)	DL Hybrid	IoT Traffic	96.2%	IoT-focused
Proposed Model	CNN-LSTM	UNSW + CIC	**97.8%**	Lightweight + Real-time

Discussion:

The proposed CNN-LSTM model achieves superior performance due to:

1. **Hybrid Learning Capability**: Combining CNN and LSTM enables both spatial and temporal feature extraction.
2. **Efficient Architecture**: Lightweight design reduces computational cost without sacrificing accuracy.
3. **Robust Generalization**: The model performs consistently across multiple datasets.

Compared to existing approaches, the proposed model provides a better trade-off between:

- Accuracy.

- Computational efficiency.
- Real-time applicability.

Conclusion and Future Work:

Conclusion:

This paper presented a lightweight hybrid deep learning model for real-time anomaly detection in network traffic environments. The proposed CNN-LSTM architecture effectively integrates spatial feature extraction and temporal sequence modeling to enhance detection performance while maintaining low computational complexity.

Experimental results on benchmark datasets demonstrated that the proposed model achieves superior performance compared to standalone CNN and LSTM models, with an accuracy of 97.8%, high precision, and low false positive rates. The confusion matrix and ROC analysis further confirmed the robustness and reliability of the model in distinguishing between normal and anomalous traffic.

In addition to performance improvements, the model maintains a compact architecture with approximately 30K parameters, making it suitable for real-time deployment in resource-constrained environments such as IoT and edge networks.

Overall, the proposed approach successfully addresses the trade-off between accuracy, efficiency, and scalability, providing a practical solution for modern intrusion detection systems.

Future Work:

Despite the promising results, **several directions can be explored to further enhance the proposed model:**

- **Real-Time Deployment Testing:** Implement and evaluate the model in real-world network environments to validate its performance under live traffic conditions.
- **Handling Imbalanced Data:** Investigate advanced techniques such as data augmentation or cost-sensitive learning to further reduce false negatives in highly imbalanced datasets.
- **Model Optimization:** Explore model compression techniques such as pruning and quantization to improve efficiency for edge devices.
- **Integration with Advanced Architectures:** Extend the model by incorporating attention mechanisms or transformer-based components to enhance temporal learning capabilities.
- **Multi-Class Classification:** Expand the model to support fine-grained attack classification instead of binary detection.
- **Explainability and Interpretability:** Apply explainable AI (XAI) techniques to improve transparency and trust in model decisions.

References:

1. Y.-C. Wang et al., "Network Anomaly Intrusion Detection Based on Deep Learning Approach," Sensors, vol. 23, no. 4, 2171, 2023.
2. H. Mennour and S. Mostefai, "Deep learning-based distributed denial-of-service detection," Int. J. Networking and Virtual Org., vol. 26, no. 1/2, pp. 80-103, 2022.
3. S. A. Khedkar et al., "Integrated Spatial and Temporal Features Based Network Intrusion Detection System Using SMOTE Sampling," IJCNIS, vol. 16, no. 2, pp. 14-27, 2024.
4. R. Ben Said et al., "CNN-BiLSTM: A Hybrid Deep Learning Approach for NIDS in SDN," IEEE Access, vol. 11, 138732, 2023.
5. S. H. Rafique et al., "Machine Learning and Deep Learning Techniques for IoT Network Anomaly Detection — Current Research Trends," Sensors, vol. 24, no. 6, 1968, 2024.
6. O. E. Elejla et al., "Deep-Learning-Based Approach to Detect ICMPv6 Flooding DDoS Attacks," Appl. Sci., vol. 12, no. 12, 6150, 2022.
7. P. Wu, H. Guo, and N. Moustafa, "Pelican: A Deep Residual Network for Network Intrusion Detection," in 50th IEEE/IFIP DSN-W, 2020.
8. K. Albulayhi and Q. Abu Al-Haija, "Adversarial Deep Learning in Anomaly based Intrusion Detection Systems," IJWMT, vol. 13, no. 4, pp. 1-10, 2023.
9. Ullah and Q. H. Mahmoud, "Design and Development of Deep Learning-Based Anomaly Detection for IoT Networks," IEEE Access, vol. 9, 103906-103926, 2021.
10. G. Andresini et al., "Multi-Channel Deep Feature Learning for Intrusion Detection," IEEE Access, vol. 8, pp. 53346-53359, 2020.