Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00339-w

Original Article

Cybercrime Forecasting Using Convolutional Neural Network-Long Short-Term Memory Networks

Parin Shah [1], Vishvendu Bhatt [1]

1. *Information Science and Technology, Asha M. Tarsadia Institute of Computer Science and Technology (AMTICS), Uka Tarsadia University, Bardoli, IND*

Received: July 21, 2026 | Review began: July 30, 2026 | Review ended: September 20, 2026 | Published: October 26, 2026

© **Copyright** 2026
This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

In today's digital world, cybercrime has evolved into a global challenge, with threats such as ransomware, data breaches, identity theft, and financial fraud affecting organizational networks and individual data. Conventional digital forensics methods primarily adopt a reactive approach in which evidence acquisition begins after a network breach has occurred. This inherent latency may increase the risk of losing critical information and can limit the effectiveness of post-incident analysis. To address this limitation, this study proposes a proactive forecasting approach for predicting the subsequent state of network activity from preceding observations. A hybrid learning architecture integrating Convolutional Neural Networks and Long Short-Term Memory networks is employed to model relationships within network-flow features and support one-step-ahead prediction. The framework is evaluated on the UNSW-NB15 benchmark and independently validated on the CIC-IDS2017 benchmark to assess its applicability beyond a single dataset. On UNSW-NB15, the model achieved an accuracy of 96.07% and a recall of 96.18%. On CIC-IDS2017, the model achieved 96.56% accuracy, 82.78% precision, 78.37% recall, an F1-score of 80.52%, and an area under the receiver operating characteristic curve of 0.9051. These findings provide additional evidence that the proposed Convolutional Neural Networks and Long Short-Term Memory forecasting framework can be applied across different network-security benchmarks, while also indicating that predictive performance varies according to dataset characteristics.

Categories: Predictive Analytics, Cyber-Physical Systems, Deep Learning

Keywords: digital forensics, predictive analytics, cybercrime forecasting, deep learning, time-series analysis, traceability, cnn-lstm, network threat forecasting

Introduction

The increasing speed and sophistication of modern cyberattacks continue to challenge the existing defense mechanisms of network organizations. Classical digital forensic approaches are mainly based on post-incident analysis, in which evidence is acquired and examined only after a cyberattack has occurred. Although this reactive approach remains legally significant, its method of operation reduces its effectiveness in real-time scenarios. By the time evidence is acquired and examined, the attackers' objectives have been achieved, and the systems are already compromised. Therefore, there is a dire need to move beyond traditional reactive methods toward proactive threat anticipation [1].

This study investigates whether sequential network traffic records contain precursor patterns that can be used to predict the subsequent network state, building on existing predictive modeling and AI-enhanced forensic approaches [2]. Rather than introducing a new Convolutional Neural Network (CNN) or Long Short-Term Memory (LSTM) architecture, the proposed approach applies these established deep learning components within a one-step-ahead forecasting

framework. The current network observation is used to estimate whether the immediately subsequent network event will correspond to normal or malicious activity. The CNN component learns localized relationships among network features, while the LSTM component processes the ordered feature representation produced by the convolutional layer.

The proposed framework is evaluated on the UNSW-NB15 dataset [3] and independently validated on the CIC-IDS2017 benchmark. Both datasets contain benign network traffic and multiple categories of malicious activity, enabling an assessment of the framework beyond a single benchmark. On UNSW-NB15, the CNN-LSTM model achieved an overall binary forecasting accuracy of 96.07%, while the CIC-IDS2017 evaluation achieved 96.56% accuracy, with a precision of 82.78%, a recall of 78.37%, an F1-score of 80.52%, and a Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) of 0.9051.

The main contributions of this study are as follows:

1) One-Step-Ahead Threat Forecasting: Network threat detection is formulated as a supervised one-step-ahead forecasting problem in which the current network-flow representation is used to predict the immediately subsequent network state. This distinguishes the proposed formulation from approaches focused primarily on classifying the current network observation.

2) Joint Spatial-Temporal Representation Learning: A hybrid CNN-LSTM framework is employed in which the 1D-CNN extracts localized relationships among network features and the LSTM models their temporal progression, reducing reliance on extensive manual feature engineering.

3) Cross-Dataset Validation: The forecasting framework is evaluated on the UNSW-NB15 benchmark and independently validated on the CIC-IDS2017 benchmark using held-out test partitions. Performance is assessed through accuracy, precision, recall, F1-score, confusion matrix analysis, and ROC-AUC to examine the applicability of the framework beyond a single benchmark dataset.

Materials And Methods

Related work

The integration of predictive analytics into network security has been increasingly explored and researched in recent years. The automation of large-scale log analysis is considered important, as manual analysis of huge volumes of data is no longer sufficient to handle the growing volume and complexity of network traffic.

Predictive Modeling for Proactive Cybersecurity

The increasing scale and complexity of network traffic have encouraged the use of automated predictive analytics for cybersecurity monitoring. Kadage et al. [2] highlighted the role of artificial intelligence in automating large-scale analysis, anomaly detection, and evidence-related processes, particularly where conventional manual workflows become difficult to scale. Within network security, machine learning has been widely applied to intrusion detection, while recent research has increasingly explored sequence-based and time-series approaches for identifying evolving attack behavior [4,5]. Building on these developments, the present study focuses primarily on proactive threat forecasting by examining whether preceding network observations can be used to predict the subsequent network state. Digital forensics is therefore considered a potential application context for the forecasting framework rather than the primary methodological objective.

Sequence and Time-Series Models

Time-series analysis has recently been identified as a powerful approach for identifying temporal patterns in network data. A comprehensive survey on time-series analysis for cybersecurity analytics and anomaly detection was presented by Landauer et al. [4]. Furthermore, sequence modeling for forensic log analytics and anomaly detection in evolving networks was demonstrated by Yang et al. [5]. Additionally, historical threat evaluations, such as ransomware risk prioritization by Massengale and Huff [6] and multi-tiered time-series intrusion detection for connected environments by

Yang et al. [7], demonstrate the effectiveness of temporal tracking. Lightweight network intrusion detection datasets introduced by Sufi [8], along with federated learning architectures for network cybersecurity evaluated by Sun et al. [9], further highlight the evolution of sequence modeling.

To address limitations in pure LSTM models, hybrid CNN-LSTM architectures for mixed threat detection were evaluated by Hassan et al. [10]. Modern systems also utilize incremental meta-learning for cyber-physical security [11], self-supervised log anomaly detection via pretrained transformer models [12], and specialized 1D-CNN architectures for in-vehicle networks [13]. Advanced structural modeling approaches include log sequence anomaly detection using self-attentive networks [14], decade-long reviews of time-series anomaly detection [15], and deep learning frameworks for threat intelligence [16]. Furthermore, Graph Neural Networks for smart grid anomaly localization [17], comprehensive ransomware detection surveys by Alraizza and Algarni [18], and ensemble deep learning architectures for industrial control systems by Al-Abassi et al. [19] demonstrate the breadth of deep learning in critical infrastructure. AI-enhanced evidence collection [2], few-shot weakly supervised anomaly detection [20], and time-series attack prediction surveys [4] further underscore the necessity of automation. Recent advances incorporating causal inference for network log analysis by Kobayashi et al. [21], deep learning NIDS reviews by Kwon et al. [22], time-series threat correlation datasets [8], large language models for cybersecurity evaluated by Xu et al. [23], artificial intelligence in digital forensics [24], forensic analytics in evolving networks [5], and digital forensics in the age of large language models [25] collectively show that deep sequence modeling is critical for real-time proactive analytics.

The CIC-IDS2017 benchmark provides an additional labeled network-flow dataset containing benign traffic and multiple attack scenarios for evaluating machine-learning-based network intrusion detection approaches [26].

Forensic Benchmarking

An important benchmark was established by Raju et al. [1] using the synthetic Digital Forensic Investigation Dataset (DFID), in which Random Forest was identified as the highest-performing traditional model, achieving an accuracy of 95%. Existing studies have demonstrated the effectiveness of time-series analysis and deep learning approaches for cybersecurity analytics, anomaly detection, and intrusion detection [4,5], while hybrid CNN-LSTM architectures have also shown strong performance in network intrusion detection tasks [10]. However, much of this work has primarily focused on detecting or classifying observed network events rather than explicitly formulating the task as one-step-ahead prediction of the subsequent network state from preceding observations. The present study addresses this distinction by applying a CNN-LSTM architecture within a one-step-ahead forecasting framework and evaluating it on two independent network-security benchmarks, UNSW-NB15 [3] and CIC-IDS2017 [26]. This cross-dataset evaluation is intended to examine whether the forecasting framework remains applicable when the underlying network features, traffic composition, and attack distribution differ across benchmark datasets. Performance is also compared with the 95% Random Forest benchmark reported by Raju et al. [1].

Theoretical background

Before the proposed method is described, a brief summary of the fundamental components of the CNN-LSTM framework is provided. This framework has been increasingly adopted in telemetry and network traffic analysis across various domains [27,28].

1D CNN

CNNs have traditionally been used for image processing tasks (2D); however, one-dimensional CNNs (1D-CNNs) have been shown to be highly effective in extracting spatial relationships within sequential data, such as adjacent network features [29-32]. In a convolutional layer, sliding filters (kernels) are applied across the input vector to generate a feature map that captures localized patterns. The convolution operation for an input sequence x and filter w is defined as

$$(x * w)_i = \sum_j x_{i-j} w_j \quad (1)$$

This process serves as an automatic feature extraction mechanism, through which underlying structural patterns of network attacks are identified from raw logs before being passed to subsequent recurrent layers.

LSTM

Traditional recurrent neural networks have a limited ability to retain long-term dependencies over extended sequences, primarily due to gradient degradation issues. This limitation is addressed by the LSTM architecture, which enables the preservation of long-range temporal relationships [33-38]. Rather than relying solely on a simple hidden state, a persistent memory cell is maintained through three gating mechanisms:

• Forget Gate: Irrelevant information is selectively removed from the memory cell.

• Input Gate: Relevant and important information from the current input is selectively fed into the memory cell.

• Output Gate: The output is generated based on the updated memory state and passed on to the following layers.

The spatial feature maps generated by the CNN are subsequently provided as input to the LSTM network. Through this integration, both spatial characteristics and temporal dependencies of network traffic are captured [39-46], enabling a comprehensive understanding of packet behavior across an evolving attack sequence.

Proposed methodology

Mathematical Formulation of Forecasting

The forensic forecasting problem is formulated as a supervised sequence-learning task. A large sequence of forensic log entries ordered by time t is represented by L. Each single entry l_t is associated with a feature vector X_t (the captured attributes) and a label y_t (Current Event: Normal/Attack).

To formulate the forecasting task, each network observation at time t is represented by a feature vector X_t and an associated binary label y_t, where $y_t = 0$ denotes Normal activity and $y_t = 1$ denotes Attack activity. The forecasting horizon is defined as one step ahead ($h = 1$), such that the target for the observation at time t is the label of the subsequent network observation:

$$y'_t = y_{t+1} \tag{2}$$

The model is trained by minimizing the binary cross-entropy loss between the predicted probability for the subsequent network state and its observed binary label:

$$\min_{\theta} \sum_{t=1}^{N} \mathcal{L}_{BCE}(f_{CNN-LSTM}(X_t;\theta), y_{t+1}) \tag{3}$$

This formulation trains the model to use the current network-flow feature representation to estimate the state of the immediately subsequent observation, thereby framing the task as one-step-ahead network-state forecasting rather than contemporaneous event classification.

Network Architecture

A highly customized PyTorch architecture was utilized to build the predictive and proactive model, specifically designed for rapid network telemetry. The precise tensor transformations through the model pipeline are structured as follows:

• **Input Layer:** After categorical encoding and standardization, each network observation was represented as a 42-dimensional feature vector. The input tensor was arranged as (B, 1, F), where $B = 128$ denotes the batch size, the single input channel represents the feature-vector channel, and $F = 42$ represents the model input dimension.

• **Spatial Layer (1D-CNN):** The input tensor (B, 1, F) is processed by a 1D-CNN with 32 output channels, a kernel size of 3, a stride of 1, and padding of 1. A ReLU activation is applied, followed by a MaxPooling1D operation with a kernel size of 2 and a stride of 2. This produces a reduced feature representation that is subsequently supplied to the recurrent layer.

• **Temporal Layer (LSTM):** The pooled convolutional representation is permuted to (B, F', 32), where F' denotes the reduced feature dimension after pooling. A two-layer stacked LSTM with 128 hidden units and a dropout rate of 0.3 processes this representation and produces a final hidden state used for binary forecasting.

• **Dense Output:** The final LSTM hidden state is passed through a fully connected layer with 64 units, followed by ReLU activation and dropout. A final linear layer maps the representation to a single output, and a Sigmoid activation converts this output into a probability of the subsequent network observation belonging to the Attack class.

Experimental setup

Dataset Selection and Validation (UNSW-NB15)

The UNSW-NB15 dataset [3] was selected to evaluate the proposed forecasting framework because it contains a combination of benign network traffic and multiple categories of malicious activity, providing a heterogeneous environment for sequence-based network analysis. The dataset was developed by researchers at the Australian Centre for Cyber Security using the IXIA PerfectStorm traffic-generation platform and includes both legitimate network behavior and synthetically generated contemporary attack scenarios [3].

An additional advantage of UNSW-NB15 for the present study is the diversity of its network attributes. The dataset provides 49 features describing basic connection characteristics, flow behavior, content information, and time-based network activity. These characteristics are particularly relevant to the proposed CNN-LSTM framework because the CNN component can learn localized relationships among network attributes, while the LSTM component can model their progression across consecutive observations. The dataset therefore provides an appropriate benchmark for investigating whether preceding network traffic patterns contain information useful for forecasting the subsequent network state.

The total processed dataset utilized in this study comprises 257,673 connection records. Of these, 75% of the data (193,255 records) were allocated for model training, while the remaining 25% (64,418 records) were reserved as an independent held-out test set. Across the entire corpus, 119,341 records represent benign background operations (Normal), while 138,332 records represent malicious activity distributed across nine distinct attack families.

For reference, Table *1* presents the attack-family distribution of the original UNSW-NB15 testing partition, which contains 82,332 records [3]. This distribution is included to illustrate the substantial imbalance among individual attack categories; the present study uses the combined 257,673-record corpus with the 75%/25% training-test partition described in "Data Partitioning and Evaluation Protocol" section.

Category	Exact Count	Forensic Characteristics
Normal	37,000	Legitimate background HTTP, SMTP, FTP traffic.
Generic	18,871	General exploits using block-cipher techniques against protocols.
Exploits	11,132	Targeting known software vulnerabilities (CVEs) and buffer overflows.
Fuzzers	6,062	Sending massive, randomized input data to force system crashes.
DoS	4,089	Denial of Service attempts aimed at resource exhaustion.
Reconnaissance	3,496	Covert probing, port scanning, and vulnerability mapping.
Analysis	677	Port scanning and spam emails.
Backdoor	583	Stealthy attempts to bypass authentication for remote access.
Shellcode	378	Malicious payload execution intended to launch command shells.
Worms	44	Self-replicating malware targeting network propagation.
Total	82,332	Complete evaluated dataset subset partition.

TABLE 1: Distribution of attack families in the original UNSW-NB15 testing partition

DoS, Denial of Service; FTP, File Transfer Protocol; HTTP, Hypertext Transfer Protocol; SMTP, Simple Mail Transfer Protocol

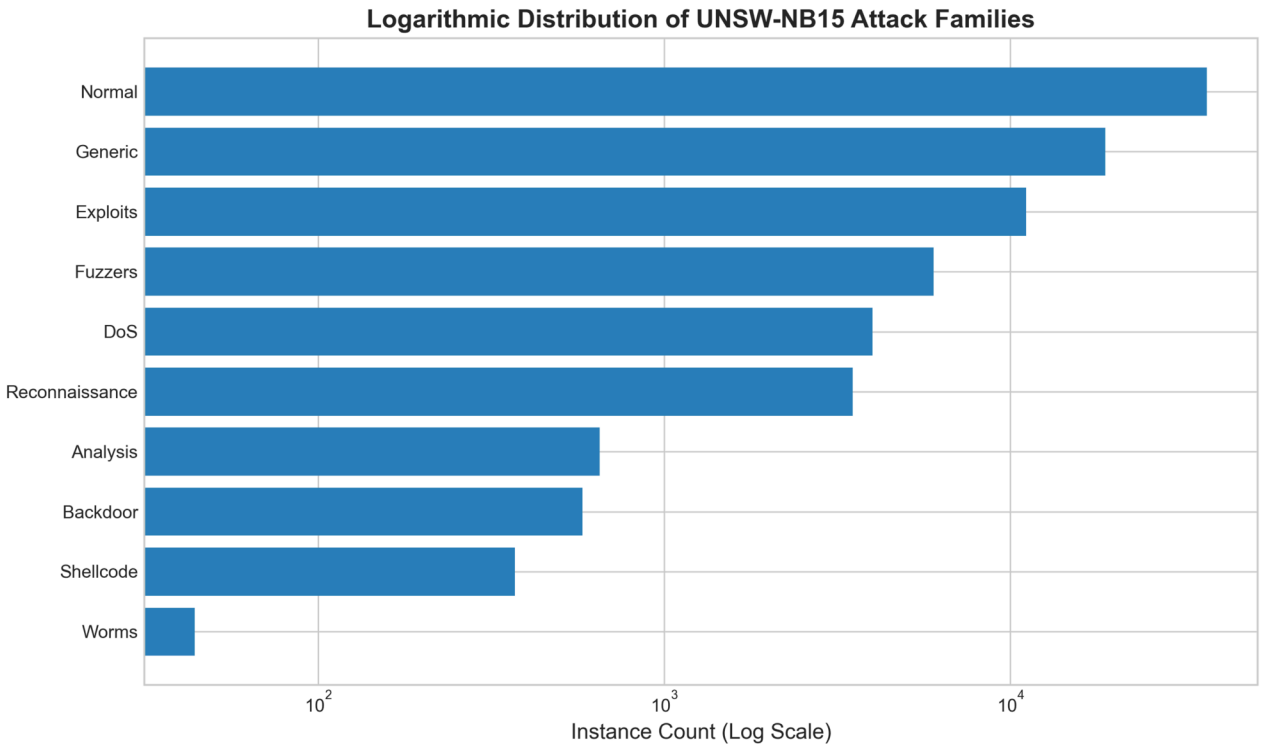

FIGURE 1: Logarithmic scale distribution illustrating the severe yet realistic class imbalance of attack families in the UNSW-NB15 dataset

DoS, Denial of Service

Table *1* and Figure *1* demonstrate substantial imbalance among the individual attack families, with some malicious categories occurring far less frequently than others. However, the forecasting task evaluated in this study is formulated as a binary Normal-versus-Attack prediction problem, in which all malicious attack families are grouped into a single Attack class. Consequently, the overall binary class distribution is considerably less imbalanced than the individual attack-family distribution. The reported accuracy, precision, recall, and F1-score therefore characterize performance at the binary level and should not be interpreted as evidence of equivalent sensitivity to every individual attack family. In particular, rare categories may contribute fewer examples to the learned representation and may require dedicated per-family evaluation in future work.

Secondary Benchmark Validation (CIC-IDS2017)

To assess the applicability of the proposed CNN-LSTM forecasting framework beyond the UNSW-NB15 benchmark, an independent evaluation was conducted using the CIC-IDS2017 benchmark dataset [26]. CIC-IDS2017 contains labeled network-flow records comprising benign traffic and multiple attack scenarios, including DDoS, PortScan, Botnet, infiltration, Web attacks, password attacks, and denial-of-service attacks. The dataset provides network-flow features extracted using CICFlowMeter and is distributed in CSV format for machine- and deep-learning applications [26].

For the present evaluation, all eight CSV files contained in the CIC-IDS2017 MachineLearningCSV collection were analyzed. The combined raw dataset comprised 2,830,743 flow records, including 2,273,097 benign records and 557,646 malicious records. During preprocessing, records containing invalid numeric values were removed, while 10 zero-variance features and one duplicate feature column were excluded, resulting in 67 active numeric network-flow features. The original attack labels were converted into a binary formulation, with BENIGN represented as 0 and all attack categories represented as 1.

The forecasting target was defined as the label of the immediately subsequent network observation ($h = 1$) within each individual capture session. The terminal observation of each session was excluded from target construction so that the forecasting target did not cross the boundary between separate capture periods. A systematic chronological stride of one in five records was then applied to reduce the computational burden while retaining the ordering of observations.

To reduce the risk of temporal information leakage, each capture session was partitioned chronologically into 60% training, 15% validation, and 25% held-out testing subsets. Feature standardization was fitted exclusively on the training partition and subsequently applied to the validation and test partitions. The final evaluation comprised 141,401 held-out test transitions.

Network Feature Set and Preprocessing

The UNSW-NB15 dataset was generated from raw PCAP (Packet Capture) traffic using Argus and Bro-IDS, resulting in 49 network features per connection record [3]. These features capture a broad range of connection, flow, content, and time-based characteristics, providing a suitable representation for spatiotemporal network analysis. For the present study, the features were grouped into four core dimensions relevant to network security:

1) Basic Packet Features: Attributes mapping foundational network connections, such as srcip (Source IP), sport (Source Port), and proto (Protocol).

2) Flow Features: Metrics utilized to evaluate the volume and duration of the network interaction, including dur (Record total duration), sbytes (Source to destination bytes), and sttl (Source to destination Time to Live).

3) Content Features: Extracted payload metadata indicating specific state changes, such as swin (Source TCP window advertisement).

4) Time-Based Features: These features are particularly relevant to network threat forecasting because they describe recent connection behavior over time. Features such as ct_srv_src (connections to the same service and source address in the last 100 connections) provide temporal information that can assist the LSTM layer in modeling changes in network behavior across consecutive observations.

Before model training, categorical attributes in the UNSW-NB15 representation were converted into numerical representations using label encoding, and the input features were standardized to provide a consistent scale for neural-network processing. After excluding the identifier and target-related fields from the UNSW-NB15 training representation, each observation used 42 input features for model training. The forecasting target was shifted by one observation ($h = 1$), such that the feature representation at time t was used to predict whether the immediately subsequent network state corresponded to Normal or Attack activity.

For CIC-IDS2017, the dataset-specific preprocessing pipeline was applied to the network-flow representation. Records containing invalid numeric values were removed, and 10 zero-variance features together with one duplicate feature were excluded, resulting in 67 active numeric flow features. The labels were converted to a binary formulation, with BENIGN represented as 0 and all attack categories represented as 1. Feature standardization was fitted exclusively on the CIC-IDS2017 training partition and then applied unchanged to the validation and held-out test partitions. The one-step-ahead target was constructed within each capture session so that the label of the subsequent network observation represented the forecasting target.

Data Partitioning and Evaluation Protocol

For UNSW-NB15, the processed dataset was partitioned into a 75% training subset and a 25% independent held-out test subset. Of the 257,673 processed records, 193,255 samples were allocated for model training and 64,418 samples were reserved for final evaluation. The held-out test subset was not used during model training and was retained for performance evaluation.

For CIC-IDS2017, a separate chronological partitioning protocol was applied within each capture session. The processed observations were divided into 60% training, 15% validation, and 25% held-out testing subsets. The validation partition was used to monitor model convergence and select the best-performing checkpoint based on validation loss, while the final test partition remained isolated until performance evaluation. Feature-standardization parameters were fitted exclusively on the training partition and then applied to the validation and test subsets.

Training Hyperparameters and Reproducibility Environment

The CNN-LSTM framework was trained using binary cross-entropy loss and the AdamW optimizer with an initial learning rate of 0.002 and a weight decay coefficient of 1×10^{-4}. Both benchmark experiments were conducted for 15 epochs with a two-layer LSTM containing 128 hidden units and a dropout rate of 0.3. A ReduceLROnPlateau learning-rate scheduler was used with a patience of two epochs and a reduction factor of 0.5. For the UNSW-NB15 experiment, a mini-batch size of 128 was used. For the CIC-IDS2017 experiment, the larger dataset was trained using a mini-batch size of 256. The CIC-IDS2017 model selection procedure used validation loss to retain the best-performing checkpoint before final evaluation on the held-out test partition. Figure *2* illustrates the training loss convergence curve of the CNN-LSTM binary cross-entropy across 15 epochs.

For reproducibility, all random number generators across Python, NumPy, and PyTorch were initialized with a fixed global random seed of 42 (torch.manual_seed(42)). The framework was implemented in Python v3.10 using PyTorch v2.1.0, CUDA v11.8, and Scikit-Learn v1.3.0. Model training and evaluation were performed on a workstation equipped with an NVIDIA GeForce RTX 3050 Laptop GPU (4 GB VRAM), an AMD Ryzen 7 4800H processor (8 cores, 16 threads), and 24 GB DDR4 RAM (3,200 MHz). The UNSW-NB15 training pipeline completed 15 epochs in approximately 22.5 minutes, while the CIC-IDS2017 experiment required approximately 27.5 minutes under the same general training configuration.

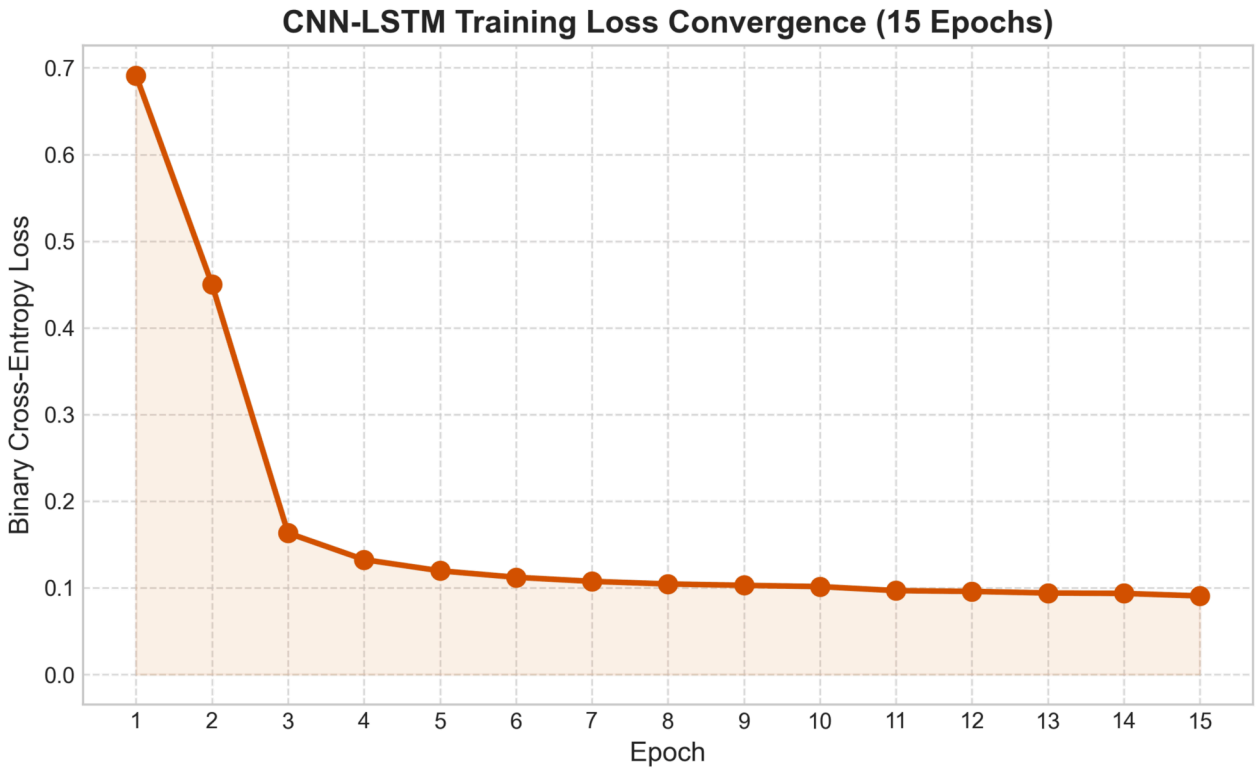

FIGURE 2: Training loss convergence curve of the CNN-LSTM binary cross-entropy across 15 epochs

CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory

Results And Discussion

Results

The primary objective of the experimental phase was to ascertain whether a deep sequence model could outperform the existing baseline established by Raju et al. [1] without relying on exhaustive manual feature engineering. The CNN-LSTM architecture was trained on the development split and subsequently evaluated against the remaining unseen 25% held-out test data (64,418 total test samples). The acquired metrics provide strong empirical support for the proposed approach within the context of network anomaly detection.

General Predictive Efficacy

An overall predictive accuracy of 96.07% was achieved by the CNN-LSTM forecasting model on the held-out UNSW-NB15 test set. Because the dataset contains heterogeneous attack categories with substantially different frequencies, accuracy alone does not fully characterize binary predictive performance. Therefore, precision, recall, and F1-score were also evaluated alongside accuracy (Figure *3*).

• **Precision (96.48%):** This indicates that 96.48% of the observations predicted as Attack were correctly classified, corresponding to a relatively low proportion of false-positive predictions in the held-out test set.

• **Recall (96.18%):** This indicates that 96.18% of the actual Attack observations in the held-out test set were correctly identified by the model.

• **F1-score (0.9633/96.33%):** The F1-score, defined as the harmonic mean of precision and recall, provides a combined measure of the model's ability to identify Attack observations while limiting false-positive predictions.

The resulting metrics indicate consistent binary predictive performance across accuracy, precision, recall, and F1-score.

How to cite this article:
Shah P, Bhatt V (October 26, 2026) Cybercrime Forecasting Using Convolutional Neural Network-Long Short-Term Memory Networks. Cureus J Comput Sci 3 : es44389-026-00339-w. DOI https://doi.org/10.7759/s44389-026-00339-w

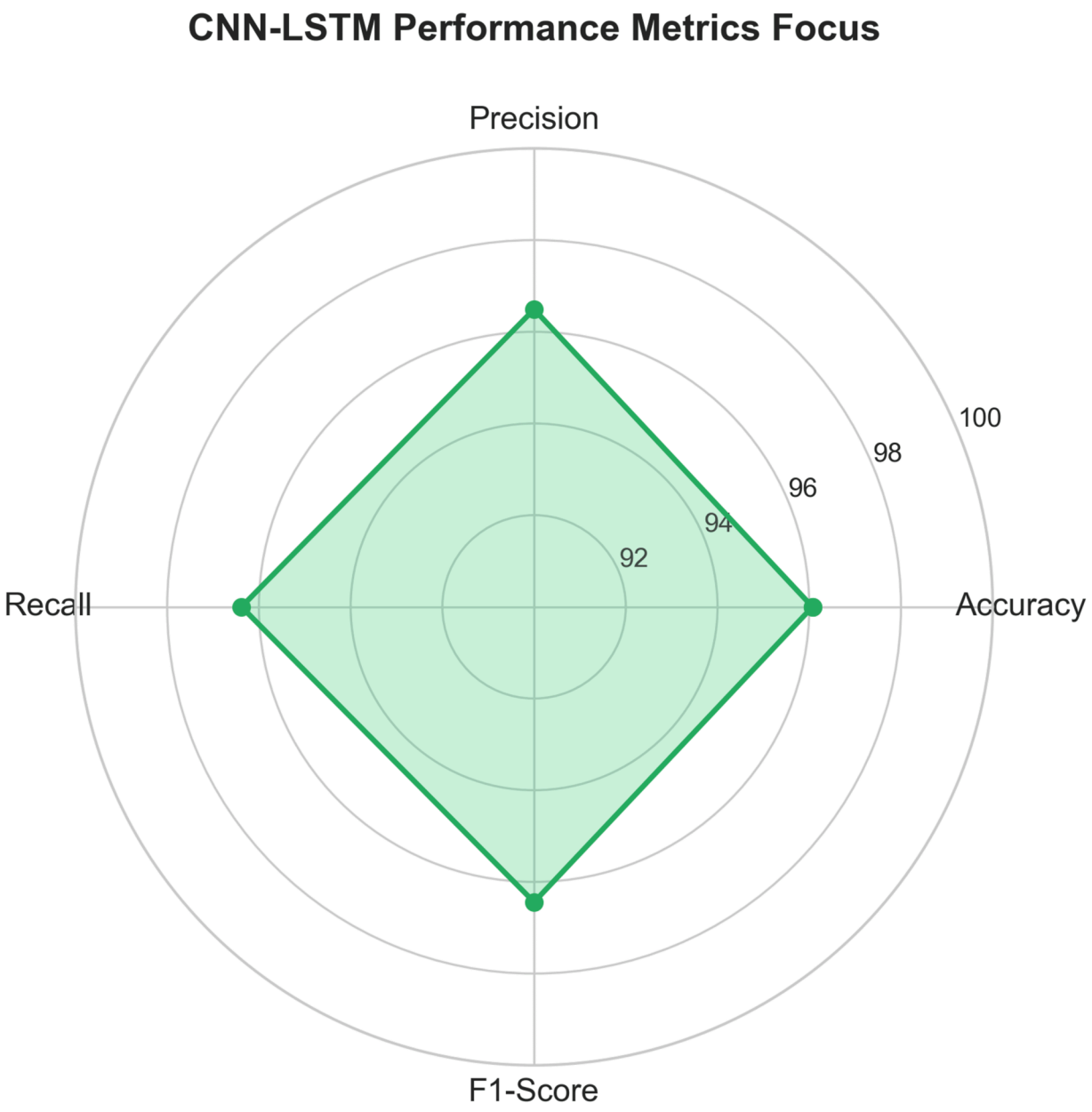

FIGURE 3: Radar chart summarizing the accuracy, precision, recall, and F1-score of the CNN-LSTM model on the UNSW-NB15 test set

CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory

Confusion Matrix Analysis

A granular breakdown of classification counts evaluated on the 64,418 unseen test samples is provided by the confusion matrix in Figure *4*. The model correctly classified 28,623 true negatives out of 29,835 benign records and 33,262 true positives out of 34,583 attack instances. A total of 1,321 false negatives were observed, corresponding to a false-negative rate of 3.82%. Additionally, 1,212 false positives were recorded, corresponding to a false-positive rate of 4.06%. These

results indicate that the model correctly distinguished the majority of benign and malicious observations in the held-out test set. The confusion matrix therefore provides a direct count-based representation of the model's binary classification performance and complements the aggregate accuracy, precision, recall, and F1-score reported in the preceding section.

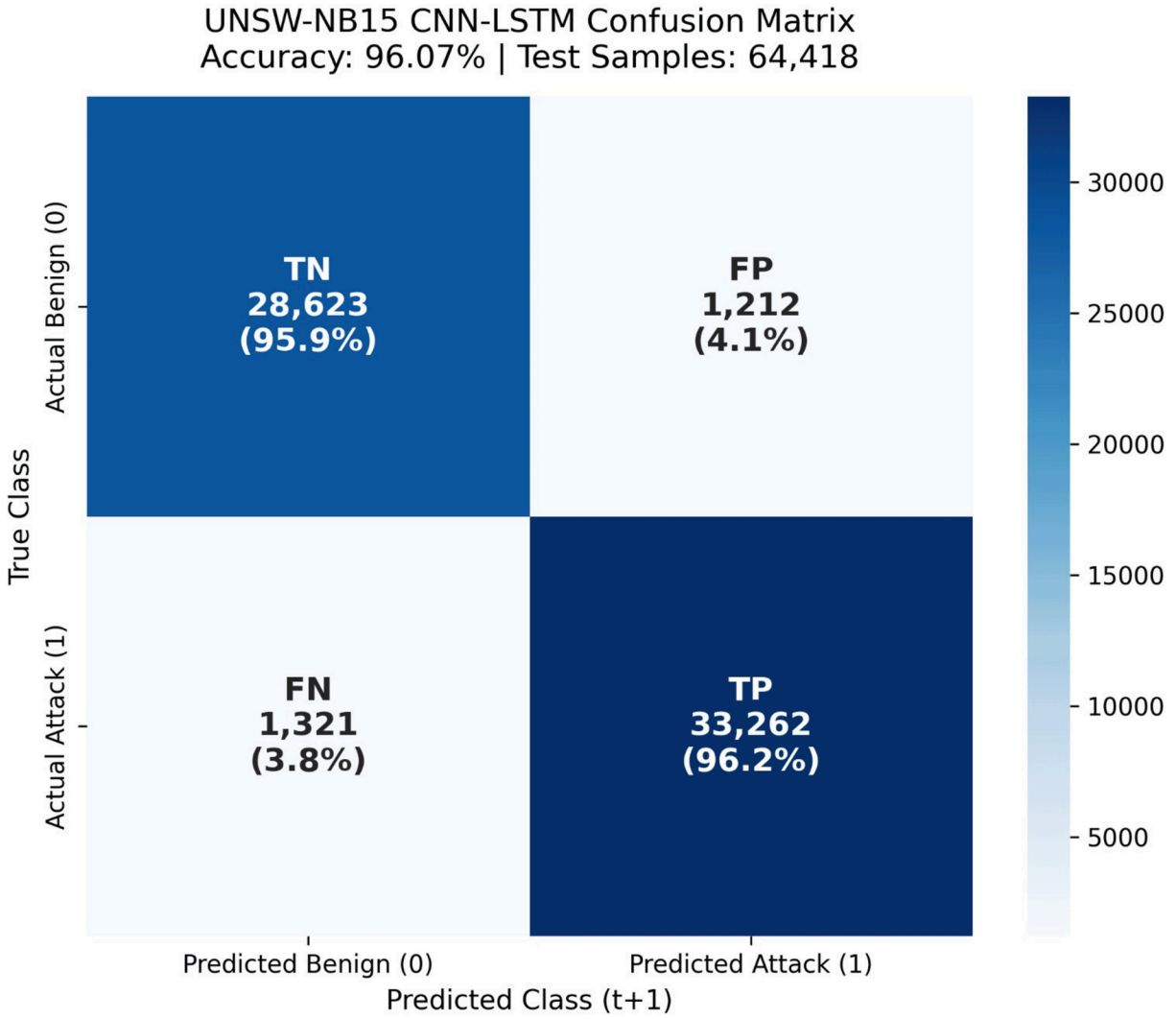

FIGURE 4: Confusion matrix of the CNN-LSTM model on the held-out UNSW-NB15 test set (64,418 samples; accuracy = 96.07%)

CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory; FN, False Negative; FP, False Positive; TN, True Negative; TP, True Positive

Discriminatory Power (ROC-AUC)

To further evaluate the discriminatory capability of the CNN-LSTM forecasting model, the ROC curve was generated using the held-out UNSW-NB15 test predictions, as illustrated in Figure *5*. The model achieved a ROC-AUC of 0.9910, indicating strong separation between the Normal and Attack classes across classification thresholds. The ROC curve shows that the model maintains a high true-positive rate while controlling the false-positive rate over a broad range of decision thresholds. These results provide complementary evidence to the accuracy, precision, recall, and F1-score reported above and indicate that the model has strong ranking capability on the held-out UNSW-NB15 test set.

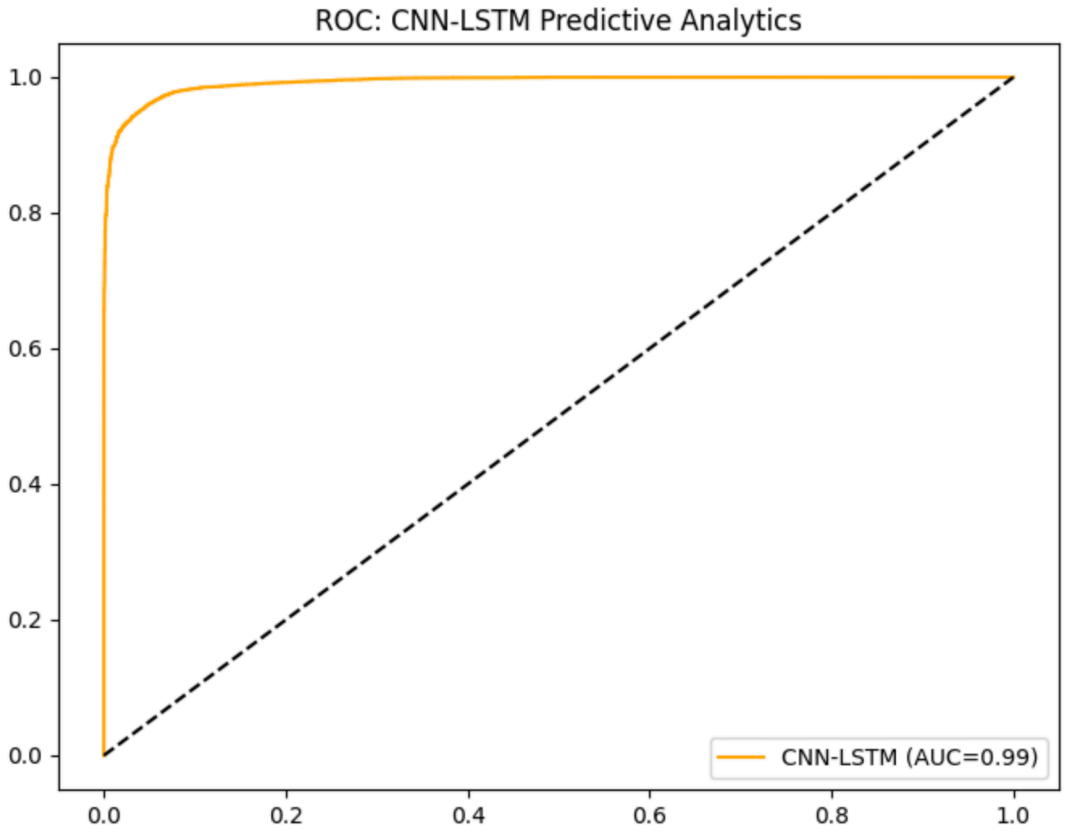

FIGURE 5: Receiver operating characteristic curve of the CNN-LSTM model on the held-out UNSW-NB15 test set (ROC-AUC = 0.9910)

CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory; ROC-AUC, Receiver Operating Characteristic-Area Under the Curve

Empirical Validation on CIC-IDS2017 Benchmark

To assess the applicability of the proposed CNN-LSTM forecasting framework beyond the UNSW-NB15 benchmark, an independent evaluation was conducted on the CIC-IDS2017 dataset. The model was evaluated on 141,401 held-out test transitions obtained through the chronological 60%/15%/25% training-validation-test protocol described in "Experimental Setup" section. The resulting accuracy was 96.56%, with a precision of 82.78%, recall of 78.37%, F1-score of 80.52%, and ROC-AUC of 0.9051.

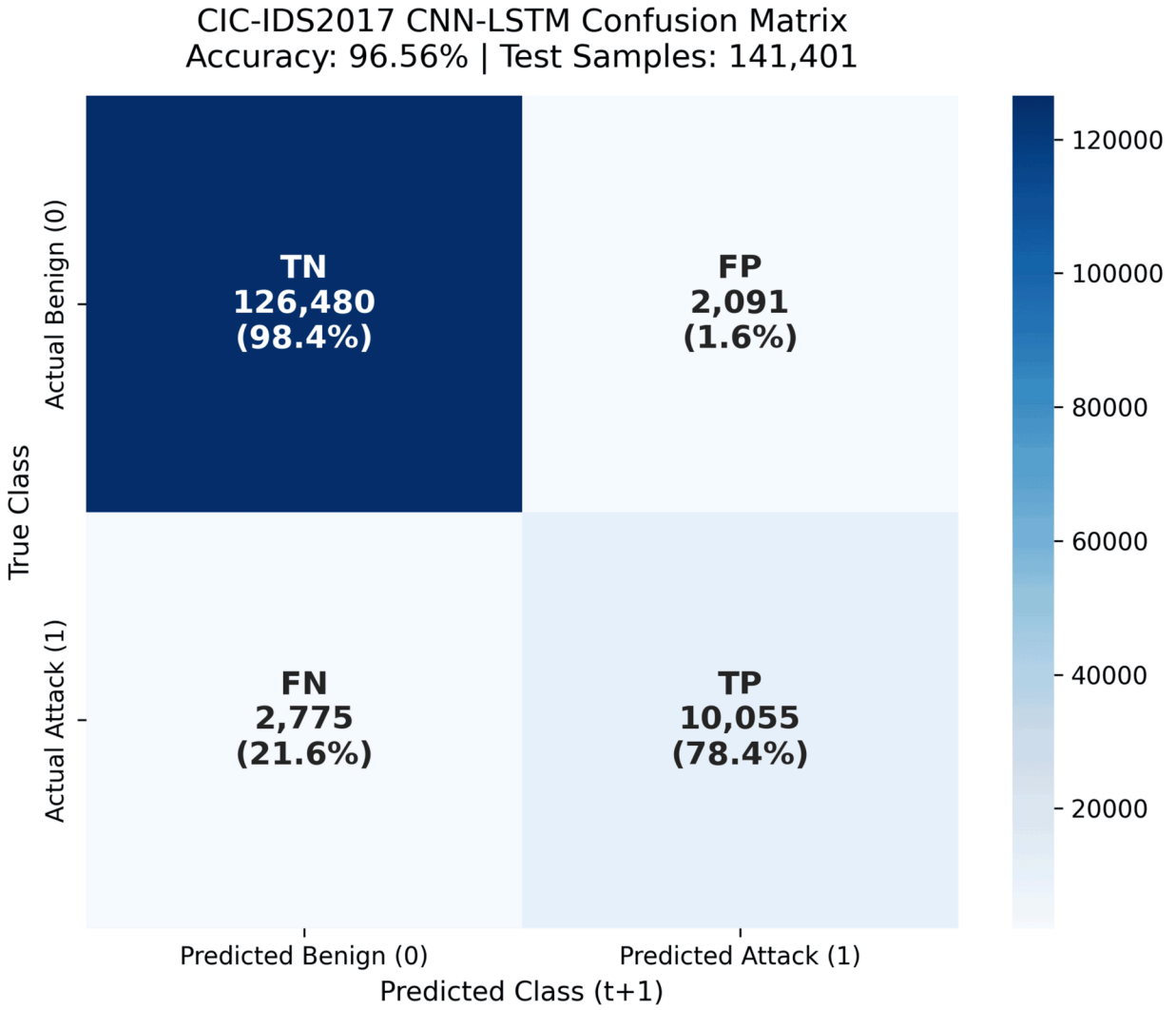

FIGURE 6: Confusion matrix of the CNN-LSTM model on the held-out CIC-IDS2017 test set (141,401 transitions; accuracy = 96.56%)

CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory; FN, False Negative; FP, False Positive; TN, True Negative; TP, True Positive

The confusion matrix in Figure *6* contains 126,480 true negatives, 2,091 false positives, 2,775 false negatives, and 10,055 true positives. The resulting false-positive rate was 1.63%, while the false-negative rate was 21.63%. These values reflect the performance of the model on the held-out CIC-IDS2017 test partition under the binary Normal-versus-Attack formulation.

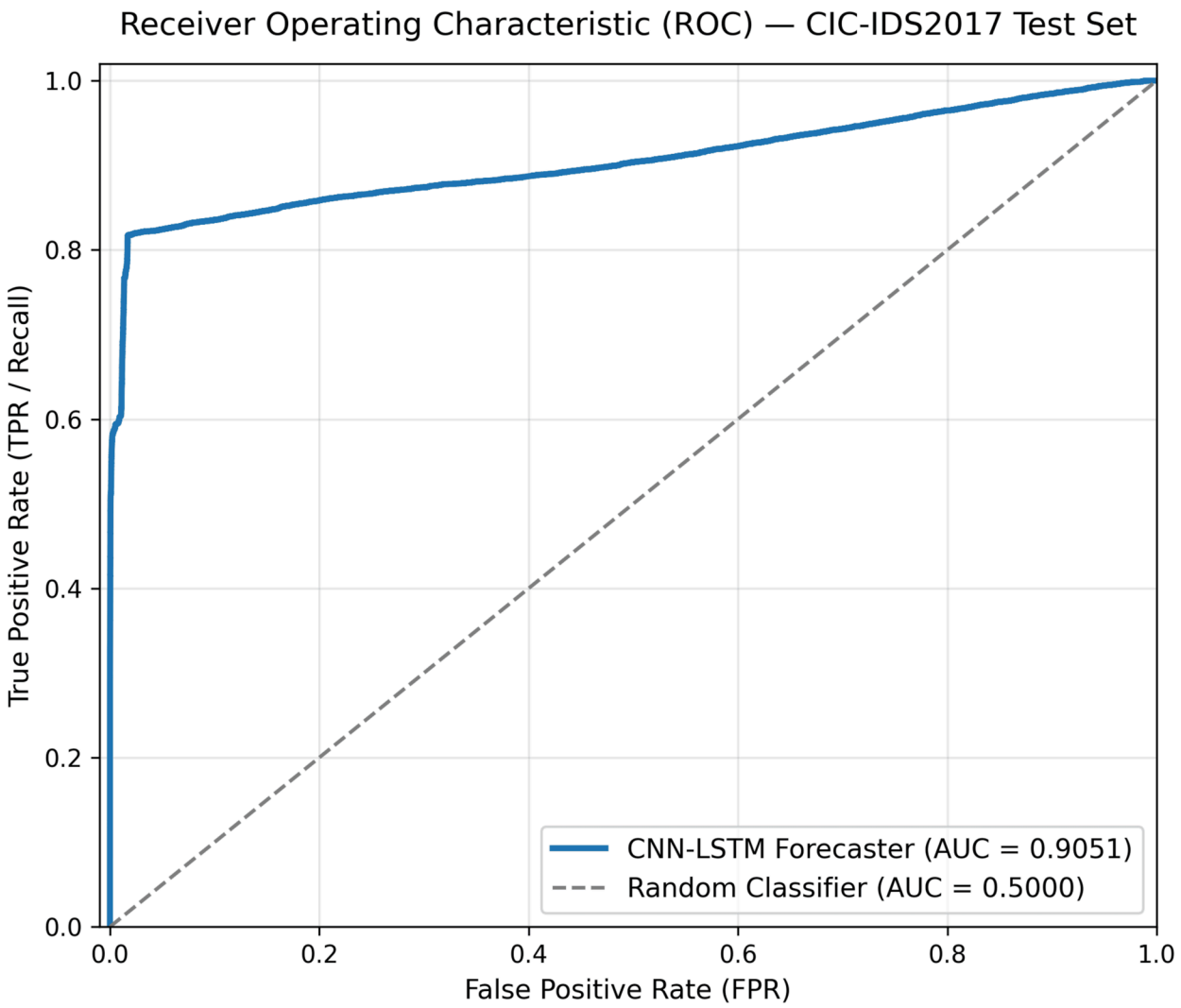

FIGURE 7: Receiver operating characteristic curve of the CNN-LSTM model on the held-out CIC-IDS2017 test set (ROC-AUC = 0.9051)

CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory; ROC-AUC, Receiver Operating Characteristic-Area Under the Curve

The ROC curve shown in Figure *7* yielded an area under the curve of 0.9051, indicating good discriminatory capability across probability thresholds.

Figure *8* presents the training and validation binary cross-entropy loss over 15 epochs, with the validation trajectory remaining stable while the training loss decreased progressively. Together, these results provide an independent evaluation of the forecasting framework on a second network-security benchmark with a different feature schema and attack distribution from UNSW-NB15.

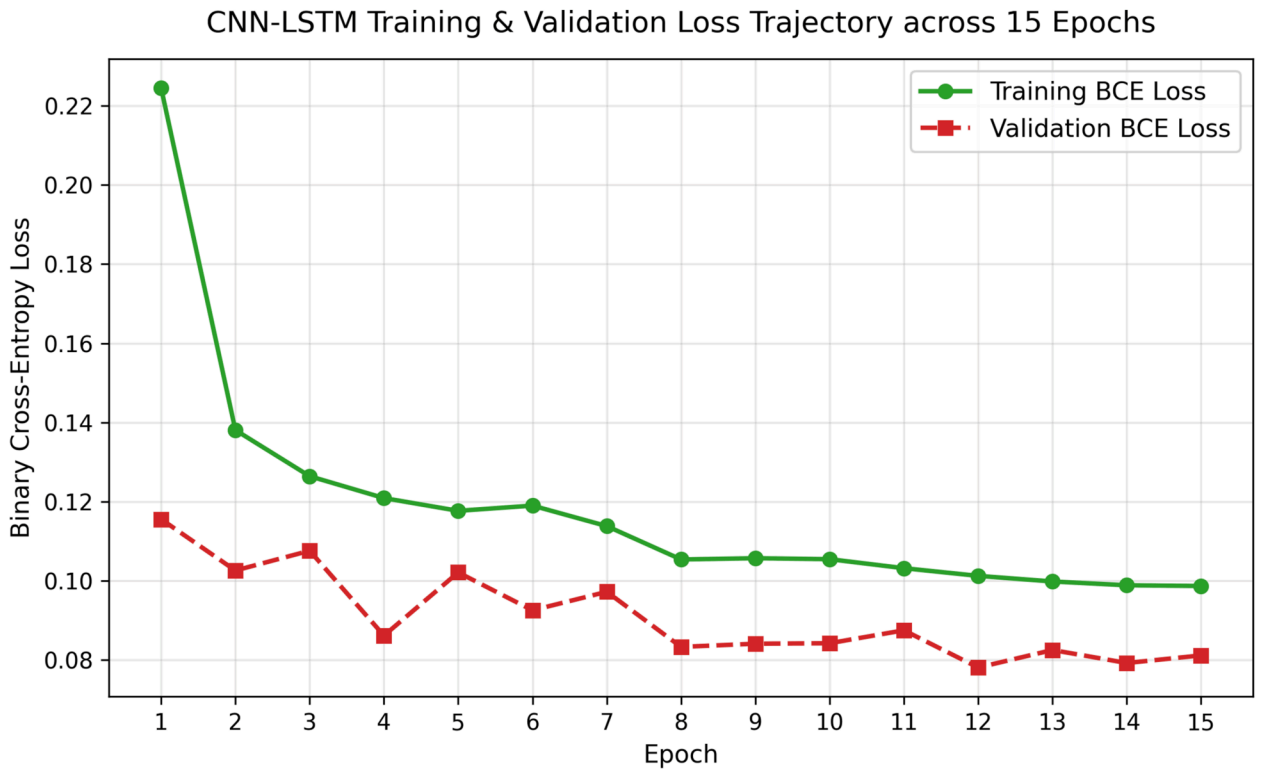

FIGURE 8: Training and validation BCE loss of the CNN-LSTM model across 15 epochs on CIC-IDS2017

BCE, Binary Cross-Entropy; CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory

Cross-Dataset Comparison

The cross-dataset results show that the CNN-LSTM forecasting framework maintained a similar overall accuracy when evaluated on the two benchmarks, while the class-sensitive metrics varied between datasets (Table *2*). The CIC-IDS2017 evaluation achieved slightly higher accuracy than UNSW-NB15 (96.56% versus 96.07%), but lower precision, recall, and F1-score. This difference indicates that predictive performance is influenced by the underlying traffic characteristics, feature distributions, and class composition of the benchmark. The independent evaluation on CIC-IDS2017 nevertheless provides additional evidence that the forecasting framework can be applied beyond the original UNSW-NB15 benchmark.

Evaluation Metric	UNSW-NB15	CIC-IDS2017
Accuracy	96.07%	96.56%
Precision	96.48%	82.78%
Recall	96.18%	78.37%
F1-score	96.33%	80.52%
ROC-AUC	0.9910	0.9051
Held-out test samples	64,418	141,401

TABLE 2: Comparison of CNN-LSTM forecasting performance across UNSW-NB15 and CIC-IDS2017

CNN-LSTM, Convolutional Neural Network-Long Short-Term Memory; ROC-AUC, Receiver Operating Characteristic-Area Under the Curve

Discussion

The experimental results provide evidence that the CNN-LSTM forecasting framework can be evaluated beyond the original UNSW-NB15 benchmark. While the UNSW-NB15 evaluation achieved 96.07% accuracy, the independent CIC-IDS2017 evaluation achieved 96.56% accuracy with lower precision, recall, and F1-score. This difference indicates that predictive performance is influenced by the characteristics of the underlying benchmark data. The following sections discuss the implications of the learned spatial representations, recurrent processing, and potential application of the forecasting framework within network security and digital forensic workflows.

The Value of Automated Feature Extraction

A notable characteristic of the proposed CNN-LSTM framework is its use of convolutional processing to learn localized relationships within the network-flow feature representation without requiring extensive manual feature construction. The UNSW-NB15 and CIC-IDS2017 experiments use different feature schemas, with the model operating on 42 input features in the UNSW-NB15 implementation and 67 active numeric flow features in the CIC-IDS2017 evaluation. Despite this difference, the same core CNN-LSTM architecture was applied across the two benchmark settings.

The results indicate that the convolutional component can learn discriminative representations from the available network-flow attributes while reducing the need to manually specify combinations of individual features. However, the present experiments do not provide direct feature-level interpretability or evidence for reliable detection of previously unseen zero-day attacks. Therefore, the results should be interpreted as evidence of automated representation learning within the evaluated benchmark datasets rather than as proof of universal signature discovery.

Temporal Representation and One-Step-Ahead Forecasting

The LSTM component provides recurrent processing of the representation produced by the convolutional layer, allowing the model to capture dependencies within the ordered feature representation used for forecasting. In the present formulation, the model is trained to predict the label of the immediately subsequent network observation ($h = 1$) rather than simply classifying the current observation. This distinction provides a forecasting-oriented formulation in which information available at the current step is used to estimate the state of the next network observation.

The cross-dataset results provide empirical support for the usefulness of this formulation. The UNSW-NB15 evaluation achieved 96.18% recall and 96.33% F1-score, while the independent CIC-IDS2017 evaluation achieved 78.37% recall and 80.52% F1-score. The lower class-sensitive metrics on CIC-IDS2017 indicate that the difficulty of one-step-ahead prediction varies with the underlying traffic distribution and attack composition. The results therefore support the use of recurrent representation learning for the evaluated forecasting task, while not establishing that specific attack stages or individual packet-level precursors can always be identified before an attack occurs.

Potential Application to Proactive Network Security

The cross-dataset results indicate the potential value of sequence-based one-step-ahead forecasting for proactive network security. Rather than limiting analysis to classification of an already observed network event, the proposed framework estimates the likelihood of the subsequent network state from the available network-flow representation. The use of sequence and time-series methods for identifying evolving cybersecurity behavior is consistent with previous research [4,5].

In a practical network-monitoring environment, the predicted probability of a subsequent malicious state could potentially be incorporated into Security Operations Center workflows as an early-warning score. When the estimated probability exceeds an operationally defined threshold, the system could support alert prioritization, additional traffic monitoring, or escalation for analyst review. Such a framework could therefore complement existing intrusion-detection mechanisms by providing information about the likely subsequent network state rather than only classifying an already observed event. However, benchmark performance alone is not sufficient to establish operational readiness. Real-world deployment would require further evaluation of inference latency, processing throughput, alert-threshold calibration, false-positive burden, changing traffic distributions, and generalization across different network environments. These operational factors were not evaluated in the present study and therefore remain important areas for future validation.

Digital forensics remains a relevant secondary application of this capability. Earlier identification of suspicious network activity could potentially support timely evidence preservation and prioritization of forensic data collection [2]. However, the present study evaluates the predictive model itself and does not experimentally assess its integration into a digital forensic workflow. Accordingly, these forensic applications are presented as potential extensions of the proposed framework rather than validated outcomes of the current study.

Comparison with reference literature

To contextualize the proposed approach, the CNN-LSTM architecture is compared against foundational benchmarks within the domain of network threat analysis.

Comparison With the DFID Benchmark

The study was partly motivated by the benchmark reported by Raju et al. [1], in which a Random Forest model achieved an accuracy of 95.00% on the synthetic DFID. The CNN-LSTM framework evaluated in the present study achieved an overall accuracy of 96.07% on the UNSW-NB15 benchmark, representing a 1.07-percentage-point difference relative to the reported DFID result.

The comparison should be interpreted in the context of the different datasets and experimental settings. DFID is a synthetic digital-forensics dataset, whereas UNSW-NB15 provides a separate network-intrusion benchmark with its own traffic characteristics and attack distribution. Therefore, the reported accuracy values should not be interpreted as a controlled head-to-head comparison. The additional evaluation on CIC-IDS2017 further examines the applicability of the CNN-LSTM framework beyond the original UNSW-NB15 benchmark.

Alignment With Previous Theoretical Findings

Previous literature has identified limitations associated with reactive cybersecurity and digital forensic workflows and has highlighted the potential value of machine-learning and time-series approaches for analyzing evolving network behavior [4,5]. The present study provides an empirical evaluation of a CNN-LSTM framework within a one-step-ahead forecasting formulation, using network-flow features to estimate the state of the immediately subsequent observation.

The results obtained on UNSW-NB15 and the independent CIC-IDS2017 benchmark provide empirical evidence that this forecasting formulation can achieve meaningful predictive performance across two network-security datasets. At the same time, the lower precision, recall, and F1-score observed on CIC-IDS2017 compared with UNSW-NB15 indicate that predictive performance is influenced by the underlying dataset characteristics and traffic distributions. These findings support further investigation of sequence-based predictive analytics while highlighting the need for broader validation across additional datasets and operational environments.

Conclusions

A proactive deep learning framework for network and information security analysis was developed and evaluated using a hybrid CNN-LSTM architecture for one-step-ahead network-state forecasting. The framework achieved an accuracy of 96.07% on the UNSW-NB15 benchmark and was independently evaluated on the CIC-IDS2017 benchmark, where it achieved 96.56% accuracy, 82.78% precision, 78.37% recall, 80.52% F1-score, and a ROC-AUC of 0.9051. The additional benchmark evaluation provides evidence that the proposed forecasting formulation can be applied beyond a single network-security dataset, while the differences in class-sensitive metrics indicate that predictive performance remains dependent on the characteristics of the underlying traffic and attack distribution.

The results support the use of CNN-LSTM-based forecasting as a potential complement to conventional reactive network-security analysis. However, the present experiments do not establish real-time deployment readiness, a fixed lead time before attack execution, or equivalent performance across all attack families and operational environments. Future work will therefore focus on validation with additional benchmark datasets and live network traffic, evaluation under changing traffic distributions and previously unseen attack patterns, and assessment of inference latency, processing throughput, decision-threshold calibration, and false-alert burden in Security Operations Center workflows. Model interpretability will also remain an important objective; attention-based or related mechanisms could help identify the network observations and attributes contributing to individual predictions and improve transparency for operational cybersecurity and digital forensic applications.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Parin Shah

Acquisition, analysis, or interpretation of data: Parin Shah, Vishvendu Bhatt

Drafting of the manuscript: Parin Shah

Critical review of the manuscript for important intellectual content: Parin Shah, Vishvendu Bhatt

Supervision: Vishvendu Bhatt

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with the ICMJE uniform disclosure form, all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted

work.

- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The data (and/or code) supporting this study are openly available in https://research.unsw.edu.au/projects/unsw-nb15-dataset (for UNSW-NB15) and https://www.unb.ca/cic/datasets/ids-2017.html (for CIC-IDS2017).

Acknowledgements

The authors used Google Gemini as an AI-assisted tool during manuscript preparation for language refinement, editorial assistance, and LaTeX formatting. The AI tool was not used to generate, fabricate, or alter experimental data, statistical results, or research findings. All analyses, experimental results, figures, interpretations, and conclusions were produced and verified by the authors, who take full responsibility for the final manuscript.

References

1. Raju KS, Ahila R, Al-Farouni M, Karuna G, Yadav K, Kumar NS: A predictive analytics approach to traceability in digital forensics using time series analysis. 2024 International Conference on IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India. 2024, 1178-1183. 10.1109/ICICAT62666.2024.10922876
2. Kadage AD, Nayak BM, Hingmire VS, Wanjale K, Bogiri N, Mandale PL: AI-Enhanced Digital Forensics: Automated Techniques for Efficient Investigation and Evidence Collection. Journal of Electrical Systems. 2024, 20:211-229. 10.52783/jes.766
3. Moustafa N, Slay J: UNSW-NB15: a comprehensive data set for network intrusion detection systems. 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia. 2015, 1-6. 10.1109/MilCIS.2015.7348942
4. Landauer M, Skopik F, Stojanovic B, Flatscher A, Ullrich T: A review of time-series analysis for cyber security analytics: from intrusion detection to attack prediction. International Journal of Information Security. 2025, 24:3. 10.1007/s10207-024-00921-0
5. Yang L, Moubayed A, Shami A, et al.: Forensic data analytics for anomaly detection in evolving networks. World Scientific Series in Digital Forensics and Cybersecurity. 2023, 99-137. 10.1142/9789811273209_0004
6. Massengale S, Huff PD: Assessing and prioritizing ransomware risk based on historical victim data. Security and Privacy in Communication Networks. Alrabaee S, Choo KKR, Damiani E, Deng RH (ed): Springer, Cham; 2026. 630:351-369. 10.1007/978-3-031-94458-1_17
7. Yang L, Moubayed A, Shami A: MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles. IEEE Internet of Things Journal. 2022, 9:616-632. 10.1109/jiot.2021.3084796
8. Sufi F: A new time series dataset for cyber-threat correlation, regression and neural-network-based forecasting. Information. 2024, 15:199. 10.3390/info15040199
9. Sun Z, Xu J, Li J, Zhang T: Privacy protection authentication protocol for consumer Internet of Things in horizontal federated learning environment. IEEE Transactions on Consumer Electronics. 2025, 71:10551-10560. 10.1109/tce.2025.3525497
10. Hassan MM, Gumaei A, Alsanad A, Alrubaian M, Fortino G: A hybrid deep learning model for efficient intrusion detection in big data environment. Information Sciences. 2020, 513:386-396. 10.1016/j.ins.2019.10.069
11. Tian J, Li M, Chen L, Wang Z: iADCPS: Time series anomaly detection for evolving cyber-physical systems via incremental meta-learning [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2504.04374
12. Liu X, Zheng S, Zou H, Yu H, Gao S: GLU-PM: A pretrained model toward generalized log understanding. IEEE Transactions on Reliability. 2026, 75:344-357. 10.1109/tr.2025.3642425
13. Song HM, Woo J, Kim HK: In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications. 2020, 21:100198. 10.1016/j.vehcom.2019.100198
14. Nedelkoski S, Bogatinovski J, Acker A, Cardoso J, Kao O: Self-attentive classification-based anomaly detection in unstructured logs. 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy. 2020, 1196-1201. 10.1109/ICDM50108.2020.00148

15. Boniol P, Liu Q, Huang M, Palpanas T, Paparrizos J: Dive into time-series anomaly detection: A decade review [PREPRINT]. arXiv. 2024, 10.48550/arXiv.2412.20512

16. Tounsi W, Rais H: A survey on technical threat intelligence in the age of sophisticated cyber attacks. Computers & Security. 2018, 72:212-233. 10.1016/j.cose.2017.09.001

17. Ahmed A, Sajan KS, Srivastava A, Wu Y: Anomaly detection, localization and classification using drifting synchrophasor data streams. IEEE Transactions on Smart Grid. 2021, 12:3570-3580. 10.1109/tsg.2021.3054375

18. Alraizza A, Algarni A: Ransomware detection using machine learning: A survey. Big Data and Cognitive Computing. 2023, 7:143. 10.3390/bdcc7030143

19. Al-Abassi A, Karimipour H, Dehghantanha A, Parizi RM: An ensemble deep learning-based cyber-attack detection in industrial control system. IEEE Access. 2020, 8:83965-83973. 10.1109/access.2020.2992249

20. Kale R, Thing VLL: Few-shot weakly-supervised cybersecurity anomaly detection. Computers & Security. 2023, 130:103194. 10.1016/j.cose.2023.103194

21. Kobayashi S, Otomo K, Fukuda K, Esaki H: Mining causality of network events in log data. IEEE Transactions on Network and Service Management. 2018, 15:53-67. 10.1109/tnsm.2017.2778096

22. Kwon D, Kim H, Kim J, Suh SC, Kim I, Kim KJ: A survey of deep learning-based network anomaly detection. Cluster Computing. 2019, 22:949-961. 10.1007/s10586-017-1117-8

23. Xu H, Wang S, Li N, et al.: Large Language Models for Cyber Security: A Systematic Literature Review. Association for Computing Machinery, New York, NY; 2025. 10.1145/3769676

24. Ganesh NSG, Venkatesh NGM, Prasad DVV: A systematic literature review on forensics in cloud, IoT, AI & blockchain. Illumination of Artificial Intelligence in Cybersecurity and Forensics. Misra S, Arumugam C (ed): Springer, Cham; 2022. 109:197-229. 10.1007/978-3-030-93453-8_9

25. Yin Z, Wang Z, Xu W, Zhuang J, Mozumder P, Smith A, Zhang W: Digital forensics in the age of large language models [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2504.02963

26. Sharafaldin I, Habibi Lashkari A, Ghorbani AA: Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy ICISSP. 2018, 1:108-116. 10.5220/0006639801080116

27. Halbouni A, Gunawan TS, Habaebi MH, Halbouni M, Kartiwi M, Ahmad R: CNN-LSTM: Hybrid deep neural network for network intrusion detection system. IEEE Access. 2022, 10:99837-99849. 10.1109/access.2022.3206425

28. Friha O, Ferrag MA, Benbouzid M, Berghout T, Kantarci B, Choo KKR: 2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT. Computers & Security. 2023, 127:103097. 10.1016/j.cose.2023.103097

29. Meliboev A, Alikhanov J, Kim W: 1D CNN based network intrusion detection with normalization on imbalanced data. 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan. 2020, 218-224. 10.1109/ICAIIC48513.2020.9064976

30. Qazi EUH, Almorjan A, Zia T: A one-dimensional convolutional neural network (1D-CNN) based deep learning system for network intrusion detection. Applied Sciences. 2022, 12:7986. 10.3390/app12167986

31. Liu C, Gu Z, Wang J: A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning. IEEE Access. 2021, 9:75729-75740. 10.1109/access.2021.3082147

32. Al-Haidari F, Al-Dahasi A: Hyperparameter optimization for 1D-CNN-based network intrusion detection using GA and PSO. Mathematics. 2023, 11:3724. 10.3390/math11173724

33. Van Houdt G, Mosquera C, Nápoles G: A review on the long short-term memory model. Artificial Intelligence Review. 2020, 53:5929-5955. 10.1007/s10462-020-09838-1

34. Graves A: Long short-term memory. Supervised Sequence Labelling with Recurrent Neural Networks. Springer, Berlin, Heidelberg; 2012. 385:37-45. 10.1007/978-3-642-24797-2_4

35. Staudemeyer RC, Morris ER: Understanding LSTM -- a tutorial into Long Short-Term Memory recurrent neural networks [PREPRINT]. arXiv. 2019, 10.48550/arXiv.1909.09586

36. Yin W, Kann K, Yu M, Schütze H: Comparative study of CNN and RNN for natural language processing [PREPRINT]. arXiv. 2017, 10.48550/arXiv.1702.01923

37. Lindemann B, Müller T, Vietz H, Jazdi N, Weyrich M: A survey on long short-term memory networks for time series prediction. Procedia CIRP. 2021, 99:650-655. 10.1016/j.procir.2021.03.088

38. Sherstinsky A: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena. 2020, 404:132306. 10.1016/j.physd.2019.132306

39. Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S: Deep learning approach for intelligent intrusion detection system. IEEE Access. 2019, 7:41525-41550. 10.1109/access.2019.2895334

40. Wang W, Sheng Y, Wang J, Zeng X, Ye X, Huang Y, Zhu M: HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access. 2018, 6:1792-1806. 10.1109/access.2017.2780250

41. Imrana Y, Xiang Y, Ali L, Abdul-Rauf Z: A bidirectional LSTM deep learning approach for intrusion detection. Expert Systems with Applications. 2021, 185:115524. 10.1016/j.eswa.2021.115524

42. Rajesh Kanna P, Santhi P: Unified deep learning approach for efficient intrusion detection system using integrated spatial-temporal features. Knowledge-Based Systems. 2021, 226:107132. 10.1016/j.knosys.2021.107132

43. Hariharan S, Annie Jerusha Y, Suganeshwari G, Syed Ibrahim SP, Tupakula U, Varadharajan V: A hybrid deep learning model for network intrusion detection system using Seq2Seq and ConvLSTM-subnets. IEEE Access. 2025, 13:30705-30721. 10.1109/access.2025.3541399

44. Roopak M, Tian GY, Chambers J: Deep learning models for cyber security in IoT networks. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA. 2019, 0452-0457. 10.1109/CCWC.2019.8666588

45. Khan MA: HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. Processes. 2021, 9:834. 10.3390/pr9050834

46. Sohi SM, Seifert JP, Ganji F: RNNIDS: Enhancing network intrusion detection systems through deep learning. Computers & Security. 2021, 102:102151. 10.1016/j.cose.2020.102151

How to cite this article:
Shah P, Bhatt V (October 26, 2026) Cybercrime Forecasting Using Convolutional Neural Network-Long Short-Term Memory Networks. Cureus J Comput Sci 3 : es44389-026-00339-w. DOI https://doi.org/10.7759/s44389-026-00339-w