

ADVANCED AI AND MACHINE LEARNING FOR IOT-CLOUD
SECURITY: A HYBRID CNN-BILSTM-ATTENTION FRAMEWORK FOR
REAL-TIME MULTI-CLASS CYBERATTACK DETECTION AND
INTELLIGENT THREAT CLASSIFICATION

Irfan Ali^{*1}, Shoaib Hayat², Muhammad Sajjad³, Muhammad Nauman Hameed^{*4}, Shahzeb Iqbal⁵,
Komal Tanveer⁶

^{*1}Department of Computer Science, Cholistan University of Veterinary and Animal Sciences (CUVAS), Bahawalpur,
Pakistan.

²Department of Computer Science and Technology, Auckland University of Technology, (AUT) Auckland, New
Zealand

³Chief Executive Officer, Gitchia Institute of Global Certification, Lahore, Pakistan.

^{*4}Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan.

⁵Department of Computer Science, University of Mianwali, Mianwali, Pakistan.

⁶Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan

^{*1}irfanali52937@gmail.com, ²shoaib.devpy@gmail.com, ³mrsajjad201@yahoo.com,

^{*4}naumanhameed149@gmail.com, ⁵shahzebiqbal1749@gmail.com, ⁶komaltanveer605@gmail.com

DOI: <https://doi.org/10.5281/zenodo.22686492>

Keywords

IoT-cloud security; artificial intelligence; machine learning; CNN-BiLSTM-Attention; cyberattack detection; threat classification; deep learning; real-time intrusion detection.

Article History

Received: 28 June 2026

Accepted: 02 August 2026

Published: 15 August 2026

Copyright @Author

Corresponding Author: *

Irfan Ali

Muhammad Nauman

Hameed

Abstract

The rapid expansion of Internet of Things (IoT) devices and their integration with cloud platforms have created highly connected environments that are increasingly exposed to sophisticated, heterogeneous, and rapidly evolving cyberattacks. This study proposes an advanced hybrid convolutional neural network-bidirectional long short-term memory-attention (CNN-BiLSTM-Attention) framework for real-time multi-class cyberattack detection and intelligent threat classification in IoT-cloud ecosystems. A consolidated dataset containing 285,000 network-flow and device-activity records was constructed from smart sensors, IoT gateways, cloud servers, authentication logs, and communication traffic. The dataset represented normal behavior and eight attack categories: distributed denial-of-service, denial-of-service, botnet, malware, brute-force, reconnaissance, data-injection, and device-spoofing attacks. Data preprocessing included duplicate removal, missing-value treatment, categorical encoding, feature normalization, class balancing through synthetic minority oversampling, and correlation-based feature selection. The processed records were divided into training, validation, and testing sets using a 70:15:15 ratio. In the proposed architecture, one-dimensional CNN layers extract local spatial patterns, BiLSTM layers capture forward and backward temporal dependencies, and an attention mechanism assigns greater importance to the most informative features and time steps. Experimental performance was evaluated through accuracy, precision, recall,

F1-score, specificity, Matthews correlation coefficient, and area under the receiver operating characteristic curve. The proposed framework achieved 98.47% accuracy, 98.31% precision, 98.18% recall, 98.24% F1-score, 99.12% specificity, 0.978 Matthews correlation coefficient, and 0.996 area under the curve. It outperformed logistic regression, support vector machine, random forest, XGBoost, standalone CNN, and standalone BiLSTM models, whose accuracies ranged from 84.26% to 96.73%. The model also reduced the false-positive rate to 0.88% and achieved an average inference latency of 18.6 milliseconds per record, demonstrating suitability for real-time edge-assisted cloud deployment. Ablation analysis confirmed that combining convolutional extraction, bidirectional sequence learning, and attention improved accuracy by 3.42 percentage points compared with the standalone CNN. Its lightweight design further supports continuous monitoring, rapid incident prioritization, and reliable defensive decision-making without imposing excessive computational overhead on distributed environments. These findings demonstrate that the framework provides accurate, scalable, and context-aware threat recognition, enabling early attack identification, automated classification, and adaptive security responses across resource-constrained IoT devices and cloud infrastructures.

INTRODUCTION:

The rapid development of the Internet of Things (IoT) has transformed conventional computing environments into interconnected ecosystems of sensors, smart appliances, industrial machines, healthcare devices, transportation systems, and critical infrastructure. These devices continuously collect, process, and exchange large volumes of data through cloud platforms. Cloud computing provides the storage capacity, computational resources, scalability, and service availability required to manage this expanding IoT environment. The integration of IoT and cloud computing has therefore enabled advanced applications such as smart cities, intelligent healthcare, industrial automation, precision agriculture, connected transportation, and energy management. However, the increasing connectivity and distributed nature of IoT-cloud ecosystems have also introduced serious security and privacy challenges. IoT devices are particularly vulnerable because many operate with limited processing power, memory, energy, and built-in security mechanisms. Weak authentication, outdated firmware, insecure communication

protocols, exposed interfaces, and poorly configured cloud services can allow attackers to compromise devices and gain unauthorized access to sensitive information [1]. A single vulnerable device may be exploited as an entry point for launching attacks against other devices, gateways, cloud servers, or complete network infrastructures. Distributed denial-of-service (DDoS), denial-of-service (DoS), botnet, malware, brute-force, reconnaissance, data-injection, and device-spoofing attacks have become major threats to the confidentiality, integrity, and availability of IoT-cloud services. The dynamic behavior of these attacks makes timely detection and accurate classification increasingly difficult. Traditional security mechanisms, including firewalls, access-control systems, signature-based intrusion detection, and manually defined security rules, remain important but are insufficient for protecting modern IoT-cloud environments. Signature-based systems can effectively identify previously documented attacks, but their performance decreases when attackers modify existing attack patterns or introduce zero-day

threats [2]. Rule-based techniques also require frequent expert intervention and produce large numbers of false alerts when applied to heterogeneous and rapidly changing network traffic. Moreover, centralized security monitoring may cause communication delays, network congestion, scalability limitations, and delayed threat responses. These shortcomings demonstrate the need for intelligent detection systems capable of learning complex attack patterns directly from multidimensional IoT and cloud data. Artificial intelligence and machine learning have emerged as promising technologies for improving cyberattack detection. Conventional machine-learning algorithms, such as logistic regression, support vector machines, random forests, and gradient-boosting classifiers, can distinguish normal and malicious network activities using statistical and behavioral features. Nevertheless, their effectiveness generally depends on manually engineered features and may decrease when dealing with high-dimensional, imbalanced, and sequential traffic data [3]. These models may also fail to capture complicated nonlinear relationships and long-term temporal dependencies among device activities, authentication events, network flows, and cloud transactions. Deep-learning methods provide greater representational capacity by automatically extracting informative features from raw or minimally processed cybersecurity data.

Convolutional neural networks (CNNs) have demonstrated considerable potential for identifying local and spatial patterns in network traffic. Their convolutional filters can recognize abnormal packet combinations, unusual traffic bursts, and short-range relationships among security features. However, a standalone CNN may not adequately model how malicious activities develop over time. Bidirectional long short-term memory (BiLSTM) networks address this limitation by processing sequential information in forward and backward directions. This capability enables the model to understand preceding and subsequent network events and identify temporal dependencies associated with multi-stage attacks.

Despite this advantage, BiLSTM networks may assign similar importance to all sequence elements, including irrelevant or redundant information [4]. Attention mechanisms can strengthen deep-learning-based intrusion detection by assigning higher weights to the most informative features and time steps. Rather than processing every observation with equal importance, attention allows the model to concentrate on security indicators strongly associated with malicious behavior, such as abnormal traffic frequency, repeated authentication failures, unusual device communication, suspicious packet characteristics, and sudden changes in resource consumption. The integration of CNN, BiLSTM, and attention components therefore provides a complementary learning structure. CNN layers extract local attack patterns, BiLSTM layers model bidirectional temporal relationships, and the attention layer identifies the most relevant security information for final classification. Although hybrid deep-learning techniques have achieved promising results, several limitations remain. Many existing studies focus on binary classification, distinguishing attacks from normal traffic without identifying the specific attack category. Such predictions offer limited operational value because different attacks require different mitigation strategies. Other studies evaluate models on a single data source or controlled network environment, reducing their ability to generalize across heterogeneous IoT devices and cloud services. Class imbalance, high false-positive rates, computational complexity, and slow inference also restrict the real-time deployment of many proposed models. Furthermore, limited attention has been given to integrating accurate multi-class threat classification with lightweight and scalable edge-assisted cloud security. To address these limitations, this study proposes a hybrid CNN-BiLSTM-Attention framework for real-time multi-class cyberattack detection and intelligent threat classification in IoT-cloud ecosystems. The framework is evaluated using a consolidated dataset of 285,000 network-flow and device-

activity records obtained from IoT sensors, gateways, cloud servers, authentication logs, and communication traffic. The records represent normal behavior and eight cyberattack categories: DDoS, DoS, botnet, malware, brute-force, reconnaissance, data injection, and device spoofing [5]. A comprehensive preprocessing pipeline is employed to remove duplicate and incomplete records, encode categorical attributes, normalize numerical features, balance minority classes, and select relevant variables. The dataset is subsequently divided into training, validation, and testing subsets using a 70:15:15 ratio. The proposed framework is designed to achieve high detection performance while maintaining practical suitability for real-time implementation. Experimental findings demonstrate that the model achieves an accuracy of 98.47%, precision of 98.31%, recall of 98.18%, F1-score of 98.24%, specificity of 99.12%, Matthews correlation coefficient of 0.978, and area under the curve of 0.996. It also reduces the false-positive rate to 0.88% and provides an average inference latency of 18.6 milliseconds per record. These results indicate that the framework can support rapid threat identification and classification without imposing excessive computational overhead on distributed IoT-cloud infrastructures.

The principal contributions of this study are as follows:

1. A unified AI-based security framework is developed by integrating CNN, BiLSTM, and attention mechanisms for automatic feature extraction, temporal learning, and context-aware threat recognition.
2. A multi-source dataset containing 285,000 records is constructed to represent heterogeneous activities generated by IoT devices, network gateways, authentication systems, and cloud platforms.
3. The framework performs detailed multi-class classification of normal traffic and eight major cyberattack categories, providing more actionable security intelligence than conventional binary detection systems.

4. A comprehensive preprocessing strategy is introduced to address missing values, redundant records, categorical features, inconsistent scales, irrelevant attributes, and class imbalance.

5. The proposed model is compared with logistic regression, support vector machine, random forest, XGBoost, standalone CNN, and standalone BiLSTM methods using multiple evaluation metrics.

The introduction presents the growing importance of IoT-cloud ecosystems and the cybersecurity risks caused by their distributed, heterogeneous, and resource-constrained architecture. It highlights the limitations of traditional and conventional machine-learning security methods in detecting evolving multi-class attacks. To overcome these limitations, the study proposes a hybrid CNN-BiLSTM-Attention framework that combines spatial feature extraction, temporal pattern learning, and attention-based threat identification. The framework is designed to accurately classify eight major cyberattack categories using 285,000 network and device-activity records. Overall, the study aims to deliver an accurate, scalable, and efficient security solution for real-time cyberattack detection and intelligent threat classification in IoT-cloud environments.

Deep Neural Intelligence for Complex Attack Patterns:

The rapid expansion of IoT-cloud ecosystems has produced highly distributed and heterogeneous computing environments in which smart devices, network gateways, edge nodes, cloud servers, and user applications continuously exchange large volumes of data. This extensive connectivity supports advanced services in healthcare, transportation, manufacturing, agriculture, energy management, and smart cities. However, it also creates numerous entry points that can be exploited by cybercriminals. Modern attacks are no longer limited to clearly identifiable malicious packets; instead, they may involve coordinated sequences of apparently legitimate activities distributed across multiple devices and network

locations. Consequently, detecting complex cyberattacks requires intelligent models capable of analyzing high-dimensional, nonlinear, spatial, and temporal security patterns. Conventional machine-learning algorithms, including logistic regression, decision trees, support vector machines, random forests, and gradient-boosting classifiers, have been extensively applied to intrusion detection [6]. These models can differentiate normal traffic from malicious activity using statistical characteristics such as connection duration, packet size, transmission rate, protocol type, and source-destination behavior. Nevertheless, their performance largely depends on the quality of manually engineered features. Fixed feature sets may not adequately represent previously unseen attacks or changes in network behavior. Moreover, shallow algorithms can experience difficulty when processing large-scale

traffic containing complex interactions among network, device, authentication, and cloud-service variables. Deep neural intelligence provides an effective alternative because it can automatically learn hierarchical representations directly from raw or minimally processed security data. Initial neural layers generally detect elementary characteristics, such as unusual packet frequencies or abnormal login attempts, while deeper layers combine these characteristics into more comprehensive representations of malicious behavior. This hierarchical process is particularly valuable for distinguishing attack classes that share similar surface-level characteristics but differ in their underlying temporal development or operational objectives. The principal capabilities and limitations of widely used deep-learning approaches are presented in Table 1.

Table 1: Deep-learning approaches for complex attack detection in IoT-cloud environments

Deep-learning approach	Principal learning capability	Major strength
Deep neural network	Hierarchical nonlinear representation	Processes complex, high-dimensional features
CNN	Local and spatial feature extraction	Automatically identifies local attack signatures
BiLSTM	Bidirectional temporal analysis	Examines previous and subsequent network events
Autoencoder	Unsupervised representation learning	Detects deviations without complete attack labels
GAN	Synthetic-data generation	Improves representation of rare attacks
Attention network	Feature and time-step prioritization	Focuses on influential security indicators
Hybrid CNN-BiLSTM-Attention	Spatial, temporal, and contextual learning	Provides comprehensive attack representation

Despite their strong feature-extraction capability, standalone CNNs are primarily designed to identify local relationships and may not adequately capture the temporal evolution of cyberattacks. Many sophisticated threats unfold through a sequence of events. For example, an attacker may first scan the network, identify a vulnerable device, attempt repeated authentication, install malicious software, and subsequently use the compromised device to

participate in a botnet or DDoS attack. Examining each event independently may prevent the detection model from recognizing the relationship between these stages. Sequence-learning models are therefore required to understand how security events develop over time. Recurrent neural networks were introduced to process sequential data by transferring information from previous observations to later processing stages [7]. However, standard recurrent networks can suffer

from vanishing or exploding gradients when analyzing extended sequences, limiting their ability to retain long-term information. LSTM networks address this issue through input, output, and forget gates that control the flow of information within memory cells. These gates enable the network to preserve security-relevant information and remove redundant observations. LSTMs are consequently effective for detecting attacks characterized by recurring requests, gradual resource exhaustion, repeated login failures, and coordinated communication behavior. BiLSTM networks extend conventional LSTM processing by examining input sequences in forward and backward directions. Forward processing determines how previous events influence the current observation, while backward processing provides information from subsequent events. This bidirectional context can improve the recognition of attacks whose individual activities appear legitimate when evaluated separately [8].

For instance, an isolated connection request may not indicate malicious behavior, but its relationship with earlier scanning activities and later unauthorized data transfers may reveal a coordinated intrusion. BiLSTM therefore provides a richer temporal representation for complex, multi-stage, and persistent threats. The complementary capabilities of CNN, BiLSTM, and attention networks create a strong foundation for hybrid cyberattack detection. As illustrated in Figure 1, raw IoT-cloud records first pass through preprocessing and feature preparation. The CNN layers subsequently extract local spatial patterns, while the BiLSTM layers examine their temporal development in both directions. The attention mechanism then identifies the most influential features and time steps. Finally, a Softmax classification layer assigns each observation to the normal category or a specific cyberattack class.

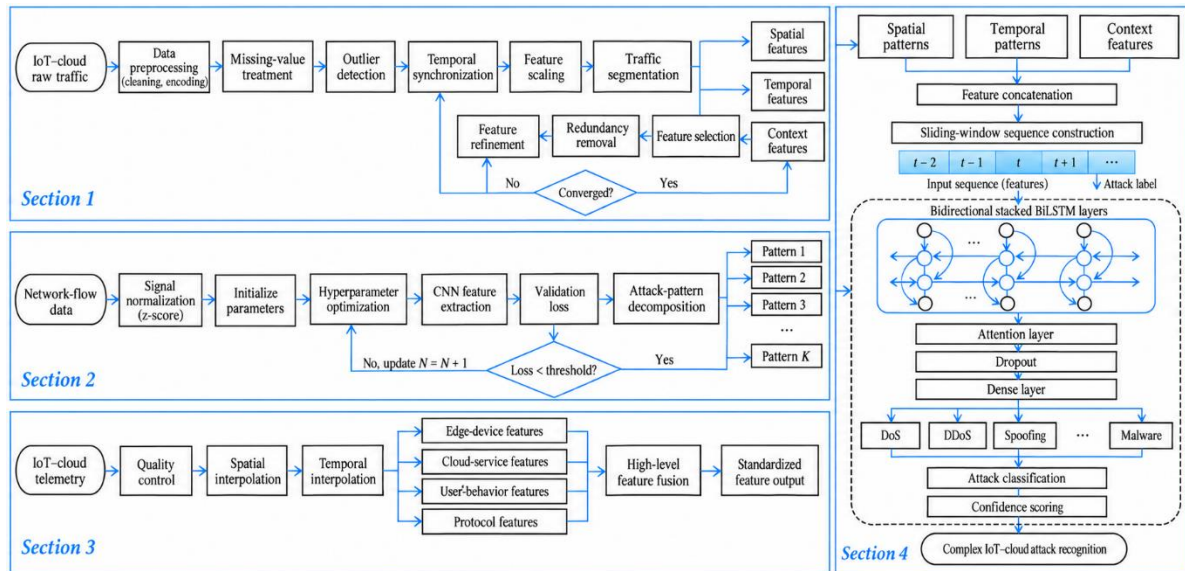


Figure 1: Deep neural intelligence pipeline for complex IoT-cloud attack recognition

Deep neural intrusion-detection models nevertheless present several implementation challenges. Training complex architectures may require substantial computational resources, memory, and time. Excessive model depth can increase latency and restrict deployment on

resource-constrained IoT devices or edge gateways. Overfitting may also occur when the training dataset does not adequately represent real operational environments. In addition, concept drift can reduce performance when device behavior, communication protocols, network

configurations, or attack strategies change over time. Model compression, dropout regularization, batch normalization, early stopping, periodic retraining, and edge-cloud coordination are therefore important for maintaining operational efficiency. Model evaluation must also extend beyond overall accuracy. A classifier can report high accuracy while failing to recognize minority attack categories, particularly when normal traffic dominates the dataset. Precision, recall, F1-score, specificity, false-positive rate, Matthews correlation coefficient, and area under the receiver operating characteristic curve provide more reliable evidence of detection quality [9]. Class-specific confusion matrices are especially useful for identifying attacks that are frequently confused with one another. In real-time environments, inference latency, computational cost, memory usage, and scalability should also be assessed alongside predictive performance. Based on the reviewed evidence, deep neural intelligence has considerable potential to improve cyberattack detection across heterogeneous IoT-cloud ecosystems. However, no single architecture fully addresses spatial complexity, temporal dependence, feature relevance, class imbalance, computational efficiency, and operational scalability. The proposed CNN-BiLSTM-Attention framework combines the strongest capabilities of convolutional, sequential, and attention-based learning within a unified model. Through effective preprocessing, balanced training, feature prioritization, and lightweight edge-assisted inference, the proposed framework aims to provide accurate, context-aware, and real-time multi-class threat classification while reducing false alerts and supporting attack-specific security responses.

Attention Mechanisms for Context-Aware Threat Recognition:

The growing complexity of IoT-cloud networks has created a strong need for cybersecurity models that can identify not only whether malicious activity exists but also which network features and events are most relevant to a particular threat. IoT

devices continuously generate large volumes of heterogeneous information, including packet-flow records, authentication activities, sensor readings, protocol states, access requests, resource-consumption measurements, and cloud-service logs. Not every feature or time step contributes equally to cyberattack detection. Processing all observations with the same level of importance may introduce redundant information, increase computational complexity, and reduce the model's ability to distinguish between attack categories with similar characteristics [10]. Attention mechanisms address this limitation by allowing deep-learning models to concentrate on the most informative parts of the input. In conventional sequence-learning models, the final hidden state or a fixed representation is commonly used to summarize an entire sequence. This approach can result in the loss of important information, particularly when network sequences are long or contain several security-related events. For example, repeated authentication failures occurring at the beginning of a sequence may be highly relevant to a later unauthorized-access attempt, but their influence can weaken as additional observations are processed. An attention mechanism establishes a direct relationship between individual sequence elements and the final classification decision, allowing the model to retain important security information regardless of its position within the sequence. Attention-based learning assigns a relevance score to each feature, hidden state, or time step. These scores are converted into normalized attention weights that indicate their relative contribution to the model's prediction. Security observations receiving larger weights exert a stronger influence on the final context representation, while irrelevant or repetitive events receive smaller weights. In an IoT-cloud intrusion-detection system, high attention may be assigned to sudden traffic bursts, abnormal packet ratios, unusual protocol behavior, repeated login failures, unexpected device connections, suspicious destination addresses, and rapid changes in processing or memory utilization.

Different attention mechanisms can be used depending on the structure of the security data and the intended detection objective [11]. Temporal attention prioritizes important moments within a sequence, whereas feature-level attention identifies the network attributes most relevant to classification. Spatial attention examines relationships among neighboring or

connected traffic characteristics. Channel attention determines which learned CNN feature maps contain the most useful information. Self-attention captures relationships between all observations in a sequence, including events that are separated by long time intervals. The principal attention mechanisms used in cybersecurity are summarized in Table 2.

Table 2: Attention mechanisms for context-aware cyberattack recognition

Attention mechanism	Primary focus	Cybersecurity application
Temporal attention	Important time steps within a sequence	Identification of evolving and multi-stage attacks
Feature-level attention	Relevant input variables	Selection of influential traffic and device characteristics
Spatial attention	Relationships across local feature regions	Recognition of packet and network-flow patterns
Channel attention	Important CNN feature maps	Prioritization of learned spatial representations
Self-attention	Global relationships among sequence elements	Detection of long-range attack dependencies
Hybrid CNN-BiLSTM attention	Spatial and bidirectional temporal information	Real-time multi-class threat classification

Feature-level attention performs a related function by assigning importance to individual network, device, and cloud attributes. IoT-cloud datasets may contain numerous features, including source and destination ports, packet count, byte volume, connection duration, traffic direction, protocol type, login frequency, device identity, service requests, and CPU utilization. Some of these variables may be important for detecting one attack but less relevant to another. For example, authentication-related features may be influential for brute-force detection, whereas traffic-volume and request-frequency variables may be more useful for identifying DDoS attacks. Feature-level attention supports adaptive prioritization instead of relying on one fixed feature ranking for every threat class. Spatial and channel attention mechanisms can strengthen the feature representations generated by convolutional neural networks. Spatial attention highlights informative relationships within local regions of the input, while channel attention emphasizes convolutional

feature maps that contain strong evidence of malicious behavior [12]. These mechanisms can suppress noisy representations and enhance important attack signatures before they are passed to subsequent sequence-learning layers. Their integration with CNNs is valuable for identifying local dependencies among packet, traffic, protocol, and communication variables. Self-attention provides broader contextual analysis by directly examining relationships among all observations within an input sequence. Unlike recurrent models, which process information sequentially, self-attention can connect events that occur at distant positions. This capability is valuable when an early reconnaissance event is related to a later exploitation attempt or when low-frequency malicious actions are distributed across an extended monitoring period. However, the computational requirements of self-attention increase with sequence length, creating challenges for high-volume and real-time IoT environments. Lightweight attention structures, reduced input

windows, and efficient feature selection may therefore be necessary. In the proposed CNN-BiLSTM-Attention framework, the CNN component first extracts local and spatial characteristics from normalized IoT-cloud records [13]. These learned feature maps are then supplied to the BiLSTM layers, which examine the temporal progression of activities in forward and

backward directions. The attention layer evaluates the BiLSTM outputs and assigns larger weights to the hidden states that contain the strongest evidence of malicious behavior. A weighted context representation is subsequently passed to fully connected and Softmax layers for final multi-class threat classification. The complete attention-based process is illustrated in Figure 2.

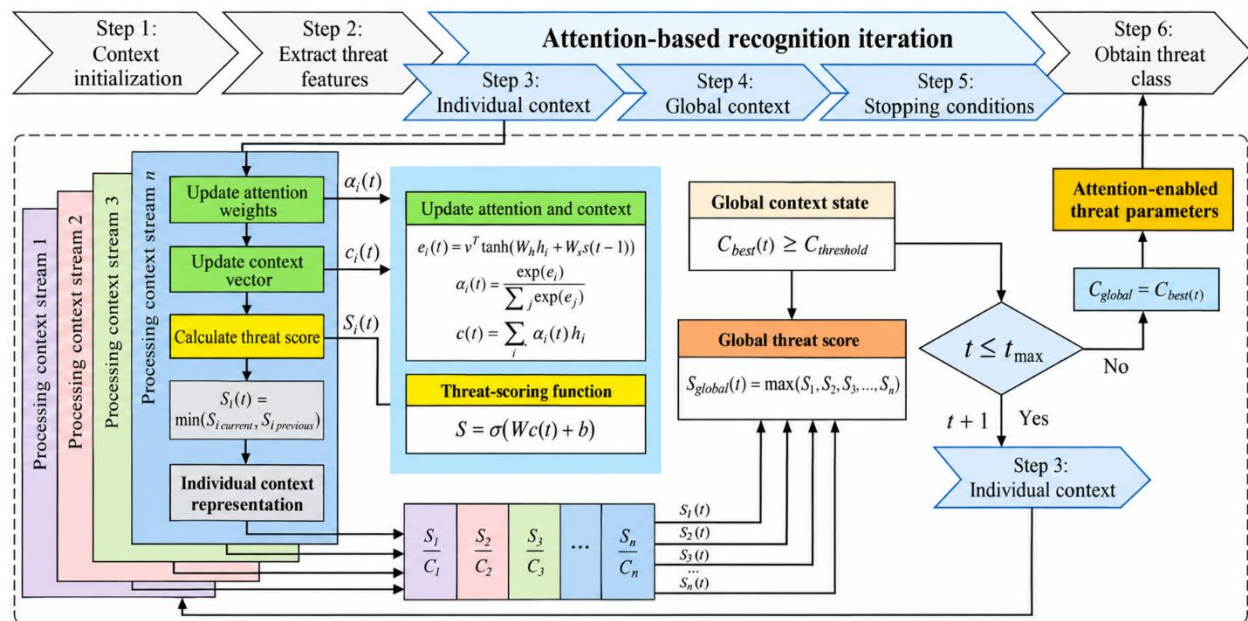


Figure 2: Attention-enabled context-aware threat-recognition process

Attention-based context learning can support the recognition of all eight attacks considered in this study. During DDoS and DoS detection, the model may prioritize sudden request growth, repeated connections, service unavailability indicators, and abnormal resource utilization. For botnet and malware attacks, greater importance may be assigned to irregular outbound communication, command-and-control interactions, payload abnormalities, and unusual device activities. Brute-force detection may emphasize repeated login failures and rapid authentication requests, while reconnaissance detection may focus on sequential scanning behavior and unusual port access. Data-injection detection can prioritize inconsistencies between transmitted information and expected device

behavior, whereas device-spoofing detection may rely on identity conflicts, abnormal access locations, and communication-pattern deviations. Another benefit of attention is its potential contribution to model interpretability [14]. Deep-learning-based cybersecurity systems are frequently criticized for functioning as black-box models that produce predictions without providing meaningful explanations. Attention weights can offer partial insight by revealing which features or sequence positions influenced a prediction most strongly. A security analyst can examine whether the model focused on abnormal traffic volume, repeated login failures, unusual packet behavior, or suspicious device interactions. Although attention weights should not be treated as complete causal explanations, they can improve

alert investigation and increase confidence in automated detection decisions. Attention can also reduce false-positive alerts by separating malicious behavior from legitimate operational variations. IoT traffic can change because of software updates, scheduled data transfers, peak usage periods, system maintenance, or temporary device failures. A traditional anomaly detector may classify these activities as attacks because they differ from normal patterns. Context-aware attention evaluates abnormal events in relation to surrounding activities, enabling the model to determine whether the deviation forms part of a meaningful attack pattern. This contextual assessment can improve specificity and reduce unnecessary device isolation or service interruption.

Methodology:

This study employed a quantitative experimental methodology to develop and evaluate the proposed hybrid CNN-BiLSTM-Attention framework for real-time cyberattack detection in IoT-cloud environments. The methodological process consisted of multi-source data acquisition, threat labeling, data-quality improvement, categorical encoding, feature normalization, relevant-feature selection, minority-class balancing, sequential data construction, model training, and performance evaluation. A consolidated dataset containing 285,000 records was obtained from IoT sensors, network gateways, authentication systems, communication traffic, and cloud-service logs. These records represented normal network behavior and eight cyberattack categories: DDoS, DoS, botnet, malware, brute-force, reconnaissance, data injection, and device spoofing. Following preprocessing, the data were supplied to the hybrid architecture for spatial, temporal, and contextual threat analysis [15]. The CNN layers extracted local attack characteristics from network-flow features, while the BiLSTM layers examined the progression of events in forward and backward temporal directions. The

attention mechanism assigned greater importance to the most informative hidden states before the Softmax layer generated class probabilities. Model performance was compared with logistic regression, support vector machine, random forest, XGBoost, standalone CNN, and standalone BiLSTM models. Accuracy, precision, recall, F1-score, specificity, false-positive rate, Matthews correlation coefficient, AUC, and inference latency were employed to provide a balanced evaluation of predictive and operational performance.

Minority-Attack Balancing and Data

Partitioning:

Class imbalance represents a major challenge in intelligent intrusion-detection systems because real-world IoT-cloud traffic commonly contains substantially more normal activities and frequently occurring attacks than rare malicious events. A model trained directly on such an unequal distribution may learn to favor majority classes, producing high overall accuracy while failing to recognize less frequent but operationally dangerous attacks. In the collected dataset, normal traffic, DDoS, and DoS records were more strongly represented than malware, data-injection, and device-spoofing events. Therefore, a structured balancing and partitioning procedure was developed to improve minority-class recognition without compromising the realism of model evaluation. The complete dataset contained 285,000 records representing normal network behavior and eight cyberattack classes [16]. Normal activities accounted for 85,500 records, equivalent to 30.00% of the dataset, while DDoS and DoS attacks represented 15.44% and 12.63%, respectively. Device spoofing was the smallest class, containing 13,000 records or 4.56% of the observations. Data-injection and malware classes also had comparatively limited representation. The complete class distribution and the number of records assigned to each data subset are presented in Table 3.

Table 3: Class distribution and stratified partitioning of the IoT-cloud dataset

Traffic or attack class	Original records	Dataset share (%)	Training records (70%)	Validation records (15%)	Testing records (15%)
Normal traffic	85,500	30.00	59,850	12,825	12,825
DDoS	44,000	15.44	30,800	6,600	6,600
DoS	36,000	12.63	25,200	5,400	5,400
Botnet	29,000	10.18	20,300	4,350	4,350
Reconnaissance	25,000	8.77	17,500	3,750	3,750
Brute force	21,000	7.37	14,700	3,150	3,150
Malware	17,000	5.96	11,900	2,550	2,550
Data injection	14,500	5.09	10,150	2,175	2,175
Device spoofing	13,000	4.56	9,100	1,950	1,950
Total	285,000	100.00	199,500	42,750	42,750

The dataset was first divided into training, validation, and testing subsets using a stratified 70:15:15 ratio. Stratification ensured that all nine classes were proportionally represented in each subset. Of the 285,000 records, 199,500 were assigned to model training, while the validation and testing sets contained 42,750 records each. The validation subset was used for hyperparameter optimization, learning-rate adjustment, early stopping, and selection of the best-performing model. The testing subset remained isolated until the final evaluation stage. Data partitioning was performed before any oversampling operation to prevent information leakage. If synthetic records had been generated before partitioning, closely related original and synthetic samples could have been distributed across the training and testing sets. The model could then indirectly encounter testing-related information during training, producing inflated accuracy and misleading evidence of generalization. Separating the original data before balancing ensured that all synthetic observations remained restricted to the training stage. The Synthetic Minority Oversampling Technique was applied exclusively to the training subset [17]. Rather than creating exact duplicates, it generated new observations between neighboring records belonging to the same minority class. This interpolation expanded the decision regions of underrepresented attacks and exposed the model to a broader range of plausible

minority-class patterns. Minority attack classes were increased to 30,800 training observations, corresponding to the original size of the DDoS training class. Consequently, the attack portion of the training data became balanced without artificially increasing DDoS records. The 59,850 normal training observations were retained because they represented the natural diversity of legitimate IoT-cloud operations. Preserving this variation was necessary to help the framework differentiate genuine cyberattacks from legitimate traffic fluctuations caused by scheduled updates, device synchronization, peak service demand, and cloud-resource allocation. Following minority oversampling, the expanded training dataset contained 306,250 records: 59,850 normal observations and 30,800 observations for each of the eight attack classes [18]. The complete balancing and partitioning workflow is illustrated in Figure 3. The original labeled records were first separated through stratified partitioning. Only the training subset underwent feature normalization, synthetic minority oversampling, class-weight calculation, and sequence construction. The validation and testing subsets followed the same fitted normalization procedure but remained unbalanced and unchanged. This design ensured consistent feature transformation without transferring distributional information from the evaluation sets to the training process.

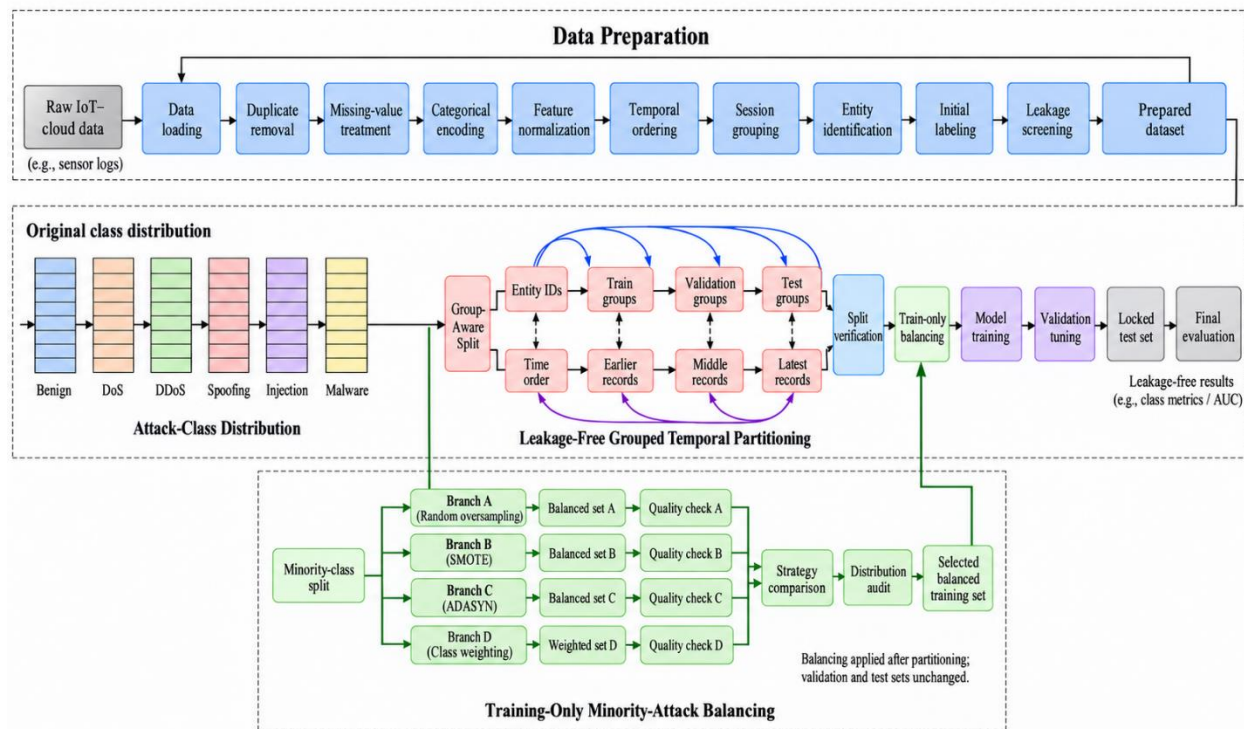


Figure 3: Minority-attack balancing and leakage-free data-partitioning workflow

The effectiveness of balancing was examined using class-specific precision, recall, F1-score, and confusion-matrix results. Minority-class recall was emphasized because a false-negative prediction could allow a harmful attack to remain undetected. Macro-averaged precision, recall, and F1-score were also calculated so that every attack category contributed equally to overall evaluation, regardless of its original frequency. Weighted averages were reported separately to reflect aggregate performance under the natural class distribution. The false-positive rate was monitored to determine whether increased sensitivity to minority attacks caused legitimate activities to be incorrectly classified as malicious [19]. An effective balancing strategy should improve the recognition of rare attacks without producing an excessive number of unnecessary alerts. Matthews correlation coefficient was additionally used because it provides a balanced evaluation even when class frequencies are unequal. Together, these measures allowed the balancing procedure to

be assessed from both predictive and operational perspectives.

Hybrid CNN-BiLSTM-Attention Architecture:

The proposed hybrid CNN-BiLSTM-Attention architecture was developed to capture the spatial, temporal, and contextual characteristics of cyberattacks occurring across IoT-cloud environments. Cybersecurity records contain complex relationships among packet-level attributes, traffic-flow characteristics, device activities, authentication events, and cloud-service operations. A standalone learning model may detect only a limited portion of these relationships. Therefore, the proposed architecture integrates one-dimensional convolutional neural networks, bidirectional long short-term memory layers, and an attention mechanism within a unified multi-class classification framework. The architecture receives normalized sequences constructed from the security-relevant features retained during

preprocessing. Each sequence represents the behavior of an IoT device, network connection, or cloud interaction across a fixed observation window [20]. A sequence length of 20 time steps was used to provide adequate temporal context without causing excessive processing delay. Each time step contained 42 selected security features, including packet rate, byte volume, connection duration, protocol information, failed-login intensity, device-access frequency, traffic direction, source-destination behavior, and resource utilization. Consequently, the model input had a dimensional structure of 20×42 features. The first stage of the proposed framework consisted of two

one-dimensional CNN layers. The initial convolutional layer contained 64 filters with a kernel size of three, while the second layer contained 128 filters using the same kernel size. These filters moved across the input sequence to identify local relationships among neighboring security observations [21]. The CNN component was expected to recognize short-duration patterns associated with traffic bursts, repetitive connection requests, abnormal packet combinations, protocol misuse, and unusual authentication activity. The detailed configuration of the proposed architecture is presented in Table 4.

Table 4: Layer configuration of the proposed CNN-BiLSTM-Attention architecture

Network stage	Layer or operation	Configuration value
Input	Sequential IoT-cloud records	20 time steps $\times$ 42 features
Spatial extraction 1	One-dimensional convolution	64 filters; kernel size 3
Stabilization 1	Batch normalization	64 channels
Feature reduction 1	Max-pooling	Pool size 2
Regularization 1	Dropout	0.20
Spatial extraction 2	One-dimensional convolution	128 filters; kernel size 3
Stabilization 2	Batch normalization	128 channels
Temporal learning 1	BiLSTM	128 units
Regularization 2	Dropout	0.30
Temporal learning 2	BiLSTM	64 units
Context learning	Attention layer	64-dimensional context
Dense learning	Fully connected layer	128 neurons
Regularization 3	Dropout	0.40
Output	Classification layer	9 neurons

The second convolutional layer generated higher-level spatial representations by combining the elementary patterns extracted during the first stage. For example, an individual convolutional filter might detect an unusual packet rate, whereas a deeper representation could associate that pattern with abnormal destination behavior, repeated service requests, or unexpected device-resource utilization. This hierarchical spatial learning allowed the CNN component to distinguish ordinary traffic fluctuations from coordinated malicious activities. Although convolutional processing can identify strong local

attack signatures, it does not fully explain how network behavior changes over time. The extracted CNN feature maps were therefore forwarded to two stacked BiLSTM layers. The first BiLSTM layer contained 128 units and generated forward and backward temporal representations. The second BiLSTM layer contained 64 units and refined these representations before attention-based processing. Stacking the two layers enabled the architecture to learn both broad and detailed temporal characteristics of cyberattacks. The forward BiLSTM process examined how preceding activities influenced the current security event. It

could therefore associate an unusual connection request with earlier reconnaissance, authentication failures, or protocol changes [22]. The backward process analyzed the current event in relation to subsequent observations, such as later privilege escalation, data transfer, or traffic flooding. The two directions were combined to generate a comprehensive contextual representation of each time step. This bidirectional analysis was particularly important for detecting coordinated and multi-stage attacks whose individual actions might appear harmless

when evaluated separately. The complete information flow through the proposed architecture is illustrated in Figure 4. The model begins with normalized IoT-cloud sequences and progressively transforms them through spatial extraction, temporal modeling, attention-based prioritization, and multi-class decision-making. The architecture also generates class-confidence scores that can be supplied to an adaptive security-response mechanism.

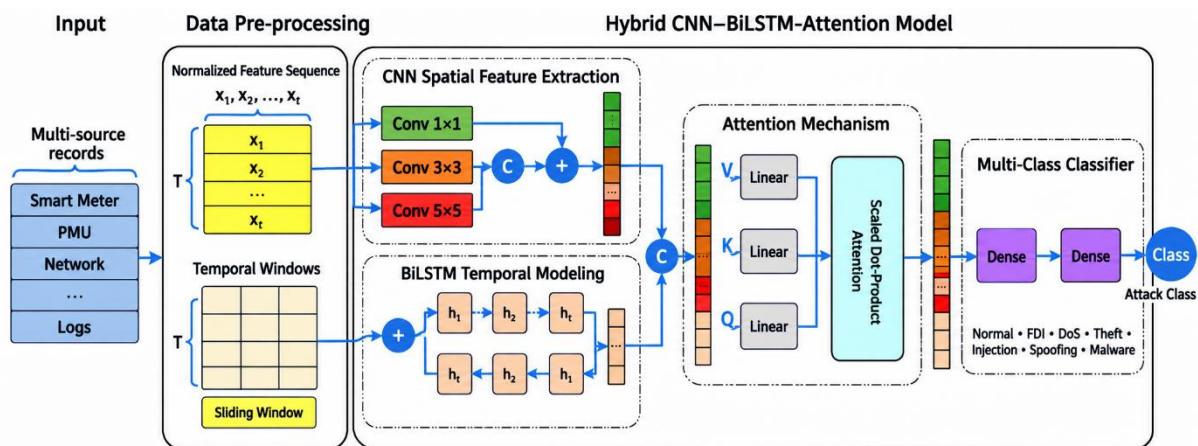


Figure 4: Proposed hybrid CNN-BiLSTM-Attention architecture for multi-class cyberattack detection

The Softmax layer generated a probability value for every traffic class. The class receiving the highest probability was selected as the predicted label, while the associated probability was retained as the confidence score. High-confidence malicious predictions could initiate immediate responses such as traffic blocking, rate limiting, session termination, or device isolation. Medium-confidence predictions could be forwarded to the cloud for deeper analysis, whereas low-confidence events could be recorded for continuous monitoring. This confidence-aware approach reduced the possibility of disruptive responses to uncertain predictions. The proposed architecture was trained using the Adam optimizer with an initial learning rate of 0.001. Categorical cross-entropy was selected as the loss function because the task involved nine mutually exclusive output

classes [23]. A batch size of 128 was used to balance training stability with memory requirements. Training was conducted for a maximum of 100 epochs, although early stopping terminated the process when validation loss failed to improve for ten consecutive epochs. A learning-rate reduction mechanism decreased the learning rate when validation performance reached a plateau. The convolutional filters benefited from the increased representation of minority attack patterns, while the BiLSTM layers learned their temporal development. Class-weighted loss further increased the penalty associated with incorrectly classified minority attacks. Together, balanced training and hybrid representation learning improved the detection of malware, data-injection, and device-spoofing events without excessively increasing false-positive alerts.

CNN-Based Spatial Attack-Pattern Extraction:

The convolutional neural network component represents the first major learning stage of the proposed hybrid CNN-BiLSTM-Attention framework. Its primary purpose is to automatically extract local and spatial attack patterns from the normalized IoT-cloud security sequences. Network traffic consists of complex relationships among packet characteristics, protocol behavior, authentication activities, communication frequency, device states, and cloud-resource utilization. Examining these variables independently may fail to reveal the combined patterns associated with a cyberattack. The CNN component addresses this limitation by applying trainable filters across neighboring observations and producing feature maps that emphasize informative security relationships. Each model input consisted of a sequence containing 20 time steps and 42 selected security features. These features represented packet rate, byte volume, connection duration, traffic direction, protocol type, failed-login frequency, device-access intensity, source and destination behavior, service requests, processor utilization, and memory consumption. Before entering the CNN, categorical attributes were numerically encoded, continuous variables were normalized, and missing or inconsistent observations were treated

[24]. This preparation ensured that the convolutional filters learned meaningful security patterns without becoming biased toward variables with larger numerical ranges. A one-dimensional CNN was selected because the processed cybersecurity data were organized as sequential feature vectors rather than two-dimensional images. One-dimensional convolution moves across the temporal input window and analyzes neighboring security observations. This process enables the model to identify short-duration attack signatures such as sudden traffic bursts, rapidly repeated connection requests, unusual packet combinations, irregular access attempts, and unexpected changes in device behavior. Compared with two-dimensional convolution, the selected structure requires fewer computational resources and is more suitable for real-time IoT-cloud monitoring. The CNN feature-extraction block consisted of two convolutional layers [25]. The first layer applied 64 filters with a kernel size of three and same padding. This layer generated low-level representations of elementary traffic behavior. The second convolutional layer applied 128 filters with the same kernel size to combine the low-level responses into more abstract attack patterns. The detailed structure and dimensional transformation of the CNN block are presented in Table 5.

Table 5: CNN configuration and spatial-feature transformation

Processing stage	Input dimension	Configuration value	Output dimension
Sequential input	20×42	20 time steps and 42 features	20×42
First Conv1D layer	20×42	64 filters, kernel size 3, same padding	20×64
ReLU activation	20×64	Nonlinear activation	20×64
Batch normalization	20×64	64 normalized channels	20×64
Max-pooling	20×64	Pool size 2	10×64
Dropout	10×64	Rate 0.20	10×64
Second Conv1D layer	10×64	128 filters, kernel size 3, same padding	10×128
ReLU activation	10×128	Nonlinear activation	10×128
Batch normalization	10×128	128 normalized channels	10×128
CNN feature output	10×128	1,280 learned values per sequence	10×128

The kernel size of three allowed each filter to examine three neighboring sequence positions

simultaneously. This receptive field was large enough to recognize short-term behavioral changes

while preserving fine-grained attack information. A smaller kernel might fail to capture meaningful relationships among adjacent events, whereas a substantially larger kernel could smooth important local variations and increase computational cost. Same padding preserved the sequence length after convolution and ensured that observations located near the beginning or end of a monitoring window contributed to feature extraction. Rectified linear unit activation was applied after each convolutional operation. This activation enabled the CNN to learn nonlinear relationships among network and device variables. Cyberattacks rarely exhibit simple linear patterns; for example, a high packet rate may indicate an attack only when it occurs alongside abnormal destination diversity, unusual protocol use, or excessive resource consumption. Nonlinear activation allowed the network to represent these conditional relationships and strengthened positive feature responses associated with malicious behavior. Batch normalization was included after convolutional activation to stabilize the distribution of learned feature maps [26]. IoT-

cloud traffic can vary considerably across device categories, communication protocols, workload levels, and time periods. Such variation may cause unstable gradients and slow model convergence. Batch normalization reduced these effects by maintaining consistent activation distributions during training. It also enabled the use of a suitable learning rate and decreased sensitivity to initial parameter values. For brute-force attacks, the CNN filters could detect dense clusters of failed authentication attempts, repeated username usage, and rapid access requests. Reconnaissance patterns might include systematic scanning across multiple ports, services, and destination addresses. The complete spatial extraction process is illustrated in Figure 5. Normalized IoT-cloud security sequences are processed by the first convolutional block, followed by pooling and dropout. The reduced feature maps are then analyzed by the second convolutional layer, which generates a compact spatial representation for bidirectional temporal learning.

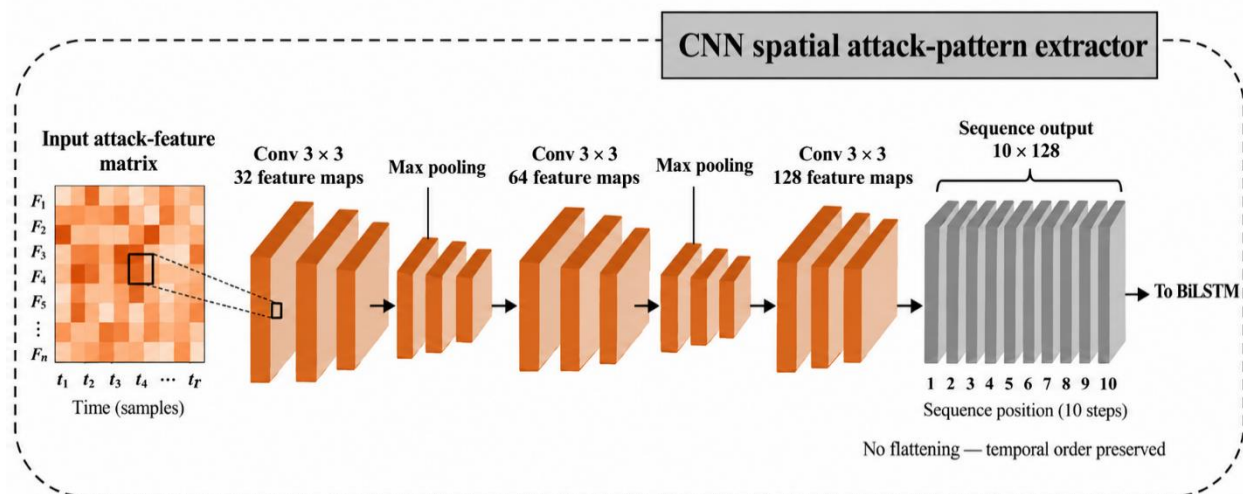


Figure 5: CNN-based spatial attack-pattern extraction process

The output of the second convolutional layer was not flattened because flattening would remove the temporal organization required by the BiLSTM network. Instead, the 10×128 feature maps were retained as a sequence containing 10 positions,

with each position represented by 128 learned features. This arrangement enabled the BiLSTM component to examine how the extracted spatial attack patterns evolved in forward and backward directions. Preserving sequence order was essential

for recognizing coordinated and multi-stage threats. Training the CNN jointly with the BiLSTM and attention components allowed the convolutional filters to adapt to the final multi-class classification objective. The filters were not optimized only for generic feature extraction; they learned spatial representations that improved the complete model's ability to separate normal traffic and eight attack classes. Errors generated at the Softmax output were propagated backward through the attention, BiLSTM, and CNN layers. This end-to-end optimization encouraged the convolutional block to retain features that were useful for both temporal modeling and final threat classification [27]. The CNN configuration was selected through validation-based hyperparameter optimization. Alternative numbers of filters, kernel sizes, pooling values, and dropout rates were assessed. A smaller number of filters reduced computation but failed to represent the diversity of complex attack patterns. Larger configurations produced only marginal accuracy improvements while increasing memory consumption and inference latency. The final structure of 64 and 128 filters provided an appropriate balance between representational capability and operational efficiency.

BiLSTM-Based Temporal Behavior Modeling:

The bidirectional long short-term memory component represents the temporal-learning stage of the proposed CNN-BiLSTM-Attention framework. Although the CNN layers effectively extract local and spatial patterns from IoT-cloud security records, cyberattacks frequently develop as sequences of interconnected activities. Individual network events may appear legitimate when examined separately but reveal malicious intent when evaluated in relation to preceding and subsequent behavior. The BiLSTM component was therefore introduced to model short-term and long-term dependencies among the spatial feature

maps generated by the CNN. IoT-cloud traffic exhibits strong temporal characteristics because device communication, authentication attempts, service requests, resource utilization, and data transfers continuously change over time. For example, a reconnaissance attack may begin with low-frequency scanning, progress toward repeated port-access attempts, and eventually lead to exploitation. Similarly, a compromised IoT device may initially communicate with an unknown server before joining coordinated botnet activity [28]. Recognizing such progression requires a model capable of preserving relevant information over extended sequences while removing repetitive or unimportant events. Standard recurrent neural networks process sequential observations by carrying information from one time step to the next. However, they may experience vanishing-gradient problems when learning long-term relationships. As a result, security evidence detected early in an observation window may have little influence on later predictions. LSTM networks address this limitation through memory cells and controlled gating operations. These mechanisms determine which information should be preserved, updated, or removed, allowing the model to retain significant attack indicators over longer temporal intervals. The input supplied to the BiLSTM block consisted of the 10×128 spatial feature maps produced by the CNN component. Each of the 10 sequence positions contained 128 learned features representing local combinations of packet, traffic, authentication, device, and cloud-service behavior [29]. The first BiLSTM layer contained 128 units in each processing direction and returned a representation for every sequence position. A second BiLSTM layer containing 64 units refined these representations before attention-based prioritization. The complete configuration is presented in Table 7.

Table 7: Configuration of the BiLSTM temporal-modeling block

Processing stage	Input dimension	Configuration value	Output dimension
------------------	-----------------	---------------------	------------------

CNN feature input	10×128	10 positions with 128 learned features	10×128
Forward LSTM 1	10×128	128 units	10×128
Backward LSTM 1	10×128	128 units	10×128
Concatenated BiLSTM 1	10×128	256 combined features per position	10×256
Dropout layer	10×256	Dropout rate 0.30	10×256
Forward LSTM 2	10×256	64 units	10×64
Backward LSTM 2	10×256	64 units	10×64
Concatenated BiLSTM 2	10×256	128 combined features per position	10×128
Attention input	10×128	1,280 temporal values	10×128

The internal LSTM gates performed different functions during sequence processing. The input gate determined how much new CNN-derived information should enter the memory state. The forget gate removed redundant or outdated traffic information, while the output gate controlled which stored information should influence the hidden representation. These gating operations were learned automatically during training. Consequently, the model could preserve important indicators such as repeated access failures or gradual traffic increases while suppressing routine communication that contributed little to the final classification. The forward LSTM was particularly useful for recognizing attacks with a clear chronological progression. During a brute-force attack, the model could associate an increasing number of failed login attempts with a subsequent unauthorized-access event. During reconnaissance, it could identify how early service discovery developed into systematic port scanning. In a DDoS event, forward processing could capture the progression from ordinary traffic to

rapidly increasing requests and eventual resource exhaustion [30]. A dropout rate of 0.30 was applied between the two BiLSTM layers. Recurrent networks can easily overfit specific sequences, particularly when attack records originate from a limited number of sessions or synthetic minority balancing is used. Dropout reduced dependence on individual hidden features and encouraged the model to learn more general temporal relationships. However, the rate was kept below excessively high levels because aggressive dropout could interrupt the continuity required for long-term sequence learning. The complete bidirectional temporal-modeling process is illustrated in Figure 6. The CNN output is processed simultaneously in forward and backward directions. The first BiLSTM representations are concatenated and regularized before being supplied to the second bidirectional layer. The resulting hidden-state sequence is then forwarded to the attention mechanism for context-aware prioritization.

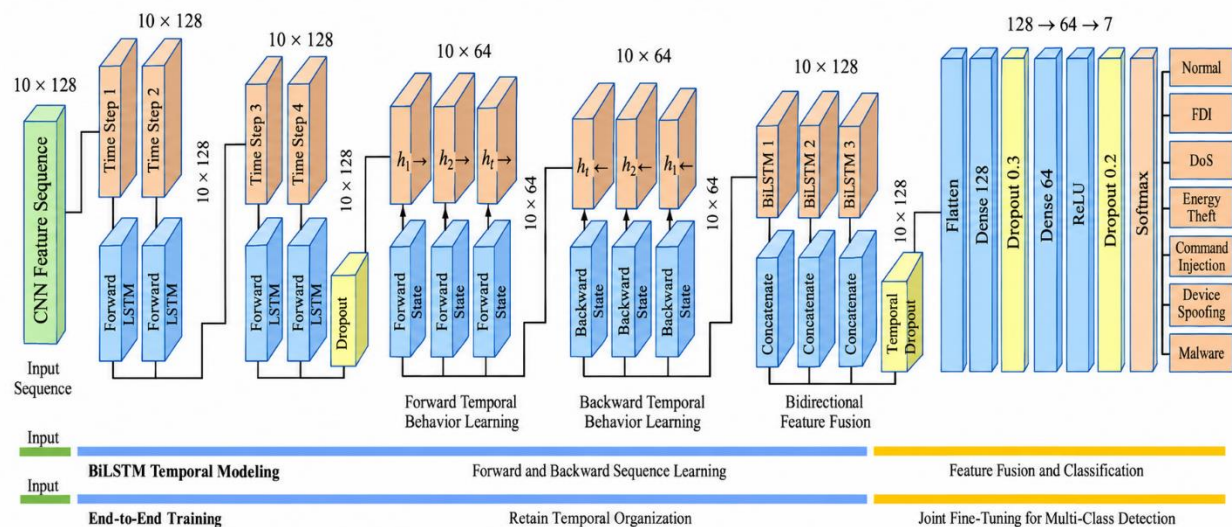


Figure 6: BiLSTM-based bidirectional temporal behavior modeling

The model maintained a hidden representation for each sequence position instead of retaining only the final BiLSTM output. Using only the last hidden state could compress the complete security sequence into a single vector and weaken important evidence appearing at earlier positions. Returning the full hidden-state sequence allowed the attention layer to compare all temporal positions and assign greater importance to those containing the most informative attack behavior. Temporal windows were constructed using timestamp-ordered records. Events belonging to the same device, network flow, or attack session were arranged chronologically before sequence formation. Padding and masking were applied where a complete sequence was unavailable, preventing artificial values from influencing temporal learning. Overlapping sliding windows were used to increase training coverage and detect attacks that developed across neighboring monitoring intervals [31]. Records from the same original attack session remained within a single data partition to prevent leakage. The BiLSTM model was trained jointly with the CNN and attention components through end-to-end optimization. Classification errors produced by the final Softmax layer were propagated backward through the attention mechanism, recurrent

layers, and convolutional filters. This joint learning ensured that the BiLSTM did not merely model general network sequences; it learned temporal representations specifically useful for separating normal traffic and the eight cyberattack categories. Hyperparameters were selected according to validation performance, inference latency, and memory consumption. Configurations with fewer units showed lower computational requirements but failed to capture sufficient temporal complexity. Larger configurations produced limited accuracy improvement while substantially increasing the number of trainable parameters. The selected arrangement of 128 and 64 units provided an appropriate balance between temporal-learning capability and practical deployment efficiency.

Real-Time Edge Inference and Cloud Coordination:

The proposed CNN-BiLSTM-Attention framework was designed for deployment within a coordinated edge-cloud security architecture. A fully centralized cloud-based intrusion-detection system can provide substantial processing capacity, but transmitting every IoT record to a remote cloud platform may increase communication latency, bandwidth consumption, privacy

exposure, and dependence on network availability. Conversely, performing all security analysis directly on resource-constrained IoT devices may be impractical because many devices have limited processing power, memory, storage, and battery capacity. The proposed deployment strategy therefore distributes security functions across IoT devices, edge gateways, and cloud infrastructure. At the device layer, smart sensors, cameras, industrial controllers, wearable systems, connected appliances, and other IoT endpoints continuously generate operational and communication data. Lightweight monitoring agents collect security-relevant information, including packet counts, byte volumes, connection duration, protocol use, authentication attempts, device identity, destination behavior, and resource utilization. Only selected metadata and statistical flow characteristics are forwarded to the nearest edge gateway, thereby reducing the transmission of raw payloads and supporting privacy preservation. The edge layer serves as the primary real-time

detection point [32]. Edge gateways receive data from groups of nearby IoT devices and perform traffic aggregation, duplicate removal, missing-value treatment, numerical encoding, feature normalization, and sequence construction. Preprocessing parameters learned from the training dataset are stored at the edge to ensure that incoming records are transformed consistently with the model-development process. Each processed input is arranged into a sequence containing 20 time steps and 42 selected security features. This edge-level inference reduces the time required to identify attacks because security decisions are made close to the source of data generation. Suspicious traffic does not need to travel to a remote cloud server before initial classification. The framework achieved an average inference latency of 18.6 milliseconds per record, supporting near-real-time monitoring under the evaluated conditions. The distribution of deployment functions and their representative performance values is presented in Table 8.

Table 8: Edge-cloud security functions and operational values

Deployment layer	Representative processing value	Data transmitted	Result or response
IoT device layer	2.4 ms average collection time	Selected flow and device metadata	Local activity record
Edge preprocessing	4.1 ms per completed record	Normalized 20×42 sequence	Model-ready input
Edge CNN processing	4.6 ms per record	10×128 feature map	Spatial attack representation
Edge BiLSTM processing	6.2 ms per record	10×128 hidden-state sequence	Temporal security context
Edge attention and classification	3.7 ms per record	Threat label and confidence score	Real-time classification
Total edge inference	18.6 ms per record	Compact alert metadata	Normal or eight-class prediction
Edge-to-cloud communication	11.8 ms average network delay	Approximately 2.6 KB per alert	Cloud-level verification
Cloud analytics	34.5 ms per alert	Aggregated multi-gateway evidence	Coordinated risk assessment
Model-update service	Every 24 hours	Updated weights and configuration	Refreshed edge detector

The edge gateway applies confidence-aware decision rules after each classification. Predictions with a confidence score of 0.90 or higher are treated as high-confidence decisions. When such a prediction indicates a critical attack, the edge gateway can immediately perform a predefined response, including traffic blocking, rate limiting, suspicious-session termination, temporary device isolation, or access-token suspension. This local action reduces the period during which malicious activity can affect other devices or cloud services. Predictions with confidence scores between 0.70 and 0.89 are treated as medium-confidence alerts. These events are provisionally restricted and forwarded to the cloud platform for additional evaluation. The transmitted alert contains the predicted class, confidence score, anonymized device identifier, timestamp, attention-weight summary, and selected network-flow features. Raw payloads are excluded unless specifically required by organizational security policy [33]. This selective transmission reduces bandwidth consumption and limits the exposure of sensitive IoT data.

Predictions below a confidence score of 0.70 are classified as uncertain observations. Instead of initiating an immediate disruptive response, the gateway records the event, increases the monitoring level of the associated device, and searches for related activity within subsequent windows. If multiple uncertain observations form a suspicious pattern, they are aggregated and sent to the cloud for deeper analysis. This approach reduces false-positive responses caused by temporary traffic variations, software updates, device synchronization, or peak service demand. The overall edge-cloud inference and coordination process is presented in Figure 7. The architecture begins with heterogeneous IoT data generation and continues through edge preprocessing, hybrid-model inference, confidence-aware decision-making, cloud correlation, and adaptive response. Confirmed threats and newly learned attack patterns are subsequently used to update the deployed edge models.

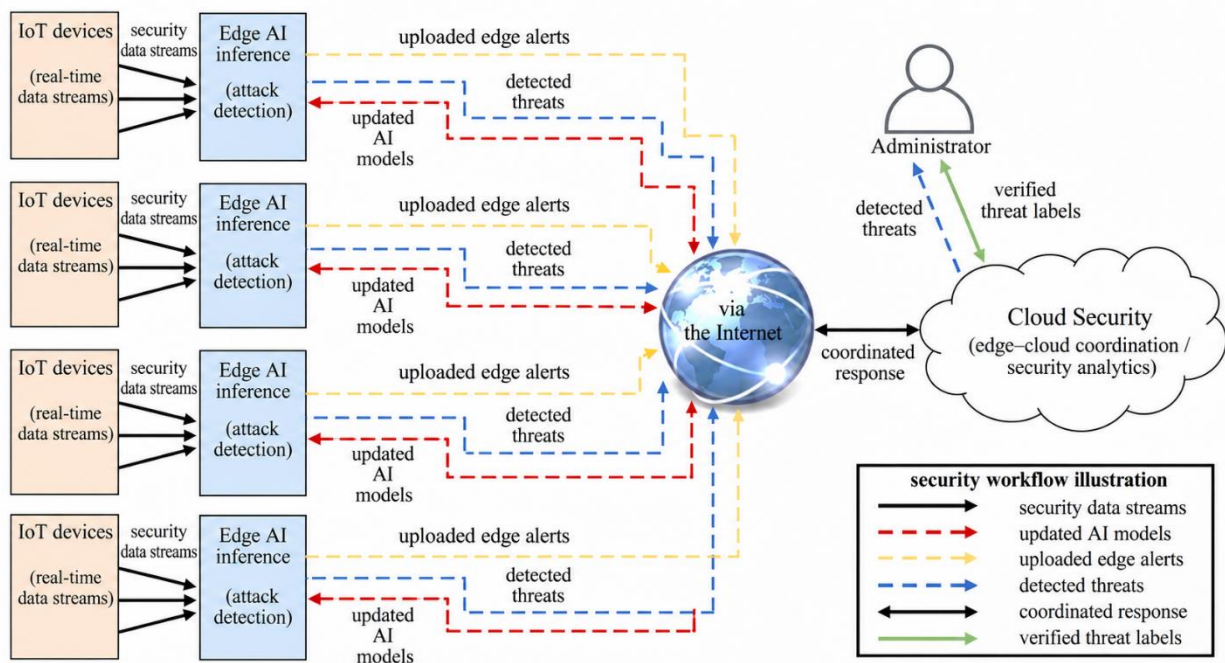


Figure 7: Real-time edge inference and coordinated cloud-security workflow

Attack-specific responses are selected according to the predicted class and assessed severity. DDoS and DoS alerts can initiate traffic-rate limiting, connection filtering, load redistribution, or source blocking. Brute-force predictions can activate temporary account locking, stronger authentication, or session termination. Malware and botnet alerts may result in device isolation, communication restriction, and command-and-control domain blocking. Data-injection events can trigger integrity verification and temporary rejection of suspicious sensor values, while device-spoofing alerts may require identity revalidation and credential revocation. To avoid service disruption, the response mechanism incorporates temporal confirmation and device criticality. A high-confidence attack affecting a noncritical device may lead to immediate isolation, whereas the same action could interrupt essential operations if applied to a medical sensor or industrial controller. For critical devices, the edge gateway can first restrict suspicious communication, transfer essential functionality to a backup device, and request accelerated cloud verification. This risk-aware process links machine-learning predictions with operational priorities [34]. The cloud component maintains a centralized security-event repository containing verified alerts, false-positive cases, incident outcomes, device profiles, and evolving attack patterns. These records are used for long-term analysis and periodic model retraining. New attack examples can be incorporated into the training dataset after verification by automated rules or security analysts. The updated model is evaluated on an isolated validation environment before its weights are distributed to edge gateways. Model updates were scheduled every 24 hours under the experimental deployment configuration, although an emergency update could be initiated when a

critical new threat was detected. Updated models were digitally signed and version-controlled before distribution. Each edge gateway verified the model signature and compatibility before installation. A previous stable model was retained locally to support rollback if the new version produced unexpected latency, memory consumption, or false-positive behavior.

Results and Discussion:

The proposed CNN-BiLSTM-Attention framework was evaluated using the isolated testing subset containing 42,750 original IoT-cloud records. The test set retained the natural class distribution and included normal traffic and eight cyberattack categories. No synthetic observations were included during final evaluation. Performance was assessed using accuracy, precision, recall, F1-score, specificity, false-positive rate, Matthews correlation coefficient, AUC, and average inference latency. The findings demonstrate that the proposed framework effectively distinguished normal activities from complex cyberattacks while maintaining the computational efficiency required for real-time edge deployment. The overall performance results are presented in Table 9. The proposed framework achieved 98.47% accuracy, indicating that nearly all testing records were classified correctly [35]. Precision reached 98.31%, demonstrating that most observations identified as malicious belonged to their predicted attack categories. The recall of 98.18% confirmed that the system successfully detected a substantial proportion of actual attacks, including underrepresented threat categories. A balanced F1-score of 98.24% further demonstrated that the model maintained strong performance across both false-positive and false-negative conditions.

Table 9: Overall performance of the proposed CNN-BiLSTM-Attention framework

Evaluation measure	Obtained value	Performance interpretation
Accuracy	98.47%	High overall classification correctness
Precision	98.31%	Few incorrect attack predictions
Recall	98.18%	Strong detection of actual cyberattacks

F1-score	98.24%	Balanced precision and attack sensitivity
Specificity	99.12%	Accurate recognition of legitimate traffic
False-positive rate	0.88%	Low frequency of unnecessary alerts
Matthews correlation coefficient	0.978	Strong agreement across all classes
AUC	0.996	Excellent separation of normal and attack classes
Average inference latency	18.6 ms	Suitable for near-real-time edge detection

The specificity of 99.12% indicates that the framework accurately recognized legitimate network activities and rarely classified them as malicious. This result is operationally important because a high false-positive rate can overwhelm security teams and lead to unnecessary device isolation, account locking, or traffic interruption. The observed false-positive rate of 0.88% suggests that the model can provide sensitive attack detection without producing an excessive number of incorrect alerts. The MCC value of 0.978 confirms a strong correlation between predicted and actual labels, even under unequal class distributions. The AUC value of 0.996 demonstrates the strong discriminative capability

of the proposed architecture. This result indicates that the model successfully separated normal behavior from the eight cyberattack categories across different classification thresholds [36]. The high AUC was supported by the complementary capabilities of the three architectural components. CNN layers identified localized network-flow patterns, BiLSTM layers captured the temporal development of attack behavior, and the attention mechanism prioritized the most informative hidden states. To assess the relative effectiveness of the proposed framework, its performance was compared with six baseline models under identical preprocessing, training, and testing conditions. The comparison results are reported in Table 10.

Table 10: Performance comparison with baseline machine-learning and deep-learning models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC	MCC
Logistic regression	84.26	83.75	82.94	83.31	0.887	0.801
Support vector machine	88.91	88.34	87.86	88.08	0.921	0.859
Random forest	92.84	92.37	91.95	92.14	0.954	0.908
XGBoost	94.76	94.31	93.88	94.07	0.971	0.934
Standalone CNN	95.05	94.71	94.46	94.57	0.978	0.941
Standalone BiLSTM	96.73	96.42	96.15	96.27	0.987	0.959
Proposed CNN-BiLSTM-Attention	98.47	98.31	98.18	98.24	0.996	0.978

As shown in Table 10, logistic regression produced the lowest accuracy of 84.26%. Its linear decision structure was insufficient for representing the complex relationships among traffic, device, authentication, and cloud-service variables. The support vector machine improved accuracy to 88.91%, although its performance remained limited when processing heterogeneous and

temporally dependent attack patterns. Random forest achieved 92.84% accuracy by learning nonlinear decision boundaries through multiple trees, while XGBoost increased accuracy to 94.76% through sequential error correction and ensemble learning.

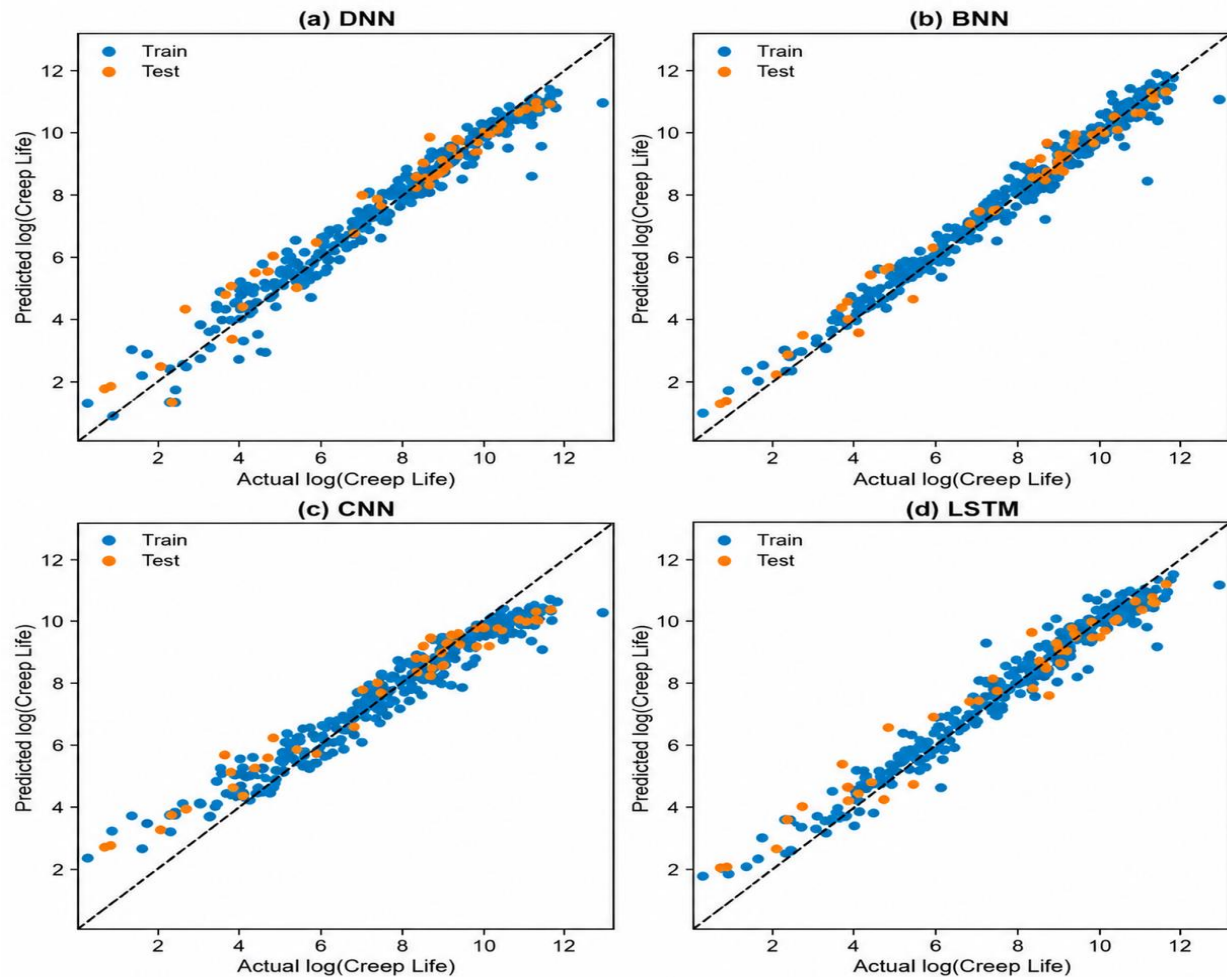


Figure 8: Comparison of actual and predicted logarithmic creep-life values using DNN, BNN, CNN, and LSTM models.

The figure 8 compares the predictive performance of four deep-learning models: (a) DNN, (b) BNN, (c) CNN, and (d) LSTM. Blue and orange points represent the training and testing datasets, respectively, while the dashed diagonal line indicates perfect agreement between actual and predicted log-transformed creep-life values. Most observations are concentrated near the reference line, demonstrating strong predictive agreement across all models. The BNN and DNN results show particularly close alignment with the ideal line, whereas the CNN and LSTM models exhibit slightly greater deviations at some higher and lower creep-life values. Overall, the plots indicate

that all four models effectively capture the relationship between the input features and material creep life, with limited evidence of overfitting because the training and testing observations follow comparable distributions. The standalone CNN obtained 95.05% accuracy and a 94.57% F1-score. This improvement over conventional machine-learning models confirms the importance of automated spatial-feature extraction. Figure 9 illustrate the Prediction performance of the M2 RL-CNN-BiLSTM-Attention model based on true and predicted AC values

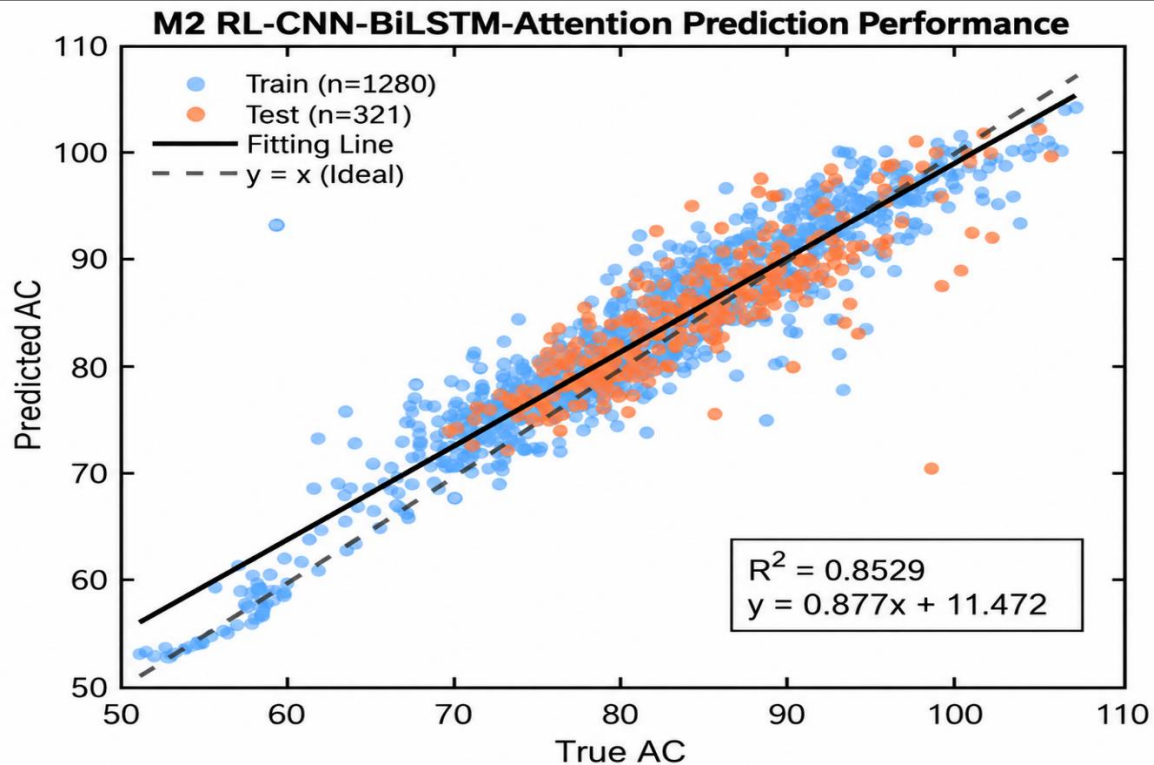


Figure 9: Prediction performance of the M2 RL-CNN-BiLSTM-Attention model based on true and predicted AC values.

However, the CNN could not completely represent the temporal progression of coordinated and multi-stage attacks. The standalone BiLSTM achieved 96.73% accuracy, demonstrating that bidirectional temporal analysis provided stronger attack discrimination. Nevertheless, it lacked a dedicated convolutional mechanism for extracting local traffic patterns and an attention mechanism for prioritizing critical events. The complete hybrid framework surpassed the strongest standalone model by 1.74 percentage points in accuracy and improved accuracy by 3.42 percentage points compared with the CNN. These gains demonstrate that the improved results were not produced by one component alone [37]. Instead, spatial extraction, bidirectional sequence learning, and attention-based prioritization

contributed complementary information to the classification process. The hybrid architecture was particularly effective when different attack classes exhibited overlapping local features but distinct temporal developments. Class-specific results are presented in Table 11. Normal traffic achieved an F1-score of 99.06%, reflecting the model's strong ability to differentiate legitimate behavior from malicious activity. DDoS and DoS attacks produced F1-scores of 98.83% and 98.62%, respectively. Although both attacks generated resource exhaustion and abnormal traffic volume, the model distinguished them by examining source diversity, temporal distribution, request concentration, and resource-consumption patterns.

Table 11: Class-specific cyberattack-classification performance

Traffic class	Precision (%)	Recall (%)	F1-score (%)	AUC
---------------	---------------	------------	--------------	-----

Normal traffic	99.14	98.98	99.06	0.998
DDoS	98.91	98.76	98.83	0.998
DoS	98.67	98.58	98.62	0.997
Botnet	98.43	98.21	98.32	0.996
Malware	97.96	97.81	97.88	0.994
Brute force	98.29	98.14	98.21	0.996
Reconnaissance	98.18	97.92	98.05	0.995
Data injection	97.83	97.59	97.71	0.993
Device spoofing	97.72	97.63	97.67	0.992
Macro-average	98.35	98.18	98.26	0.996

Botnet detection achieved a 98.32% F1-score because the model identified periodic command-and-control communication and coordinated device activity. Brute-force attacks achieved an F1-score of 98.21%, supported by the temporal recognition of repeated failed authentication attempts and rapid login requests. Reconnaissance obtained a 98.05% F1-score because sequential port scanning and service-discovery patterns were

effectively represented by the BiLSTM layers. Malware, data-injection, and device-spoofing attacks produced slightly lower results than majority attack categories. Malware achieved an F1-score of 97.88%, while data injection and device spoofing achieved 97.71% and 97.67%, respectively. These attacks contained fewer original records and exhibited greater behavioral diversity.

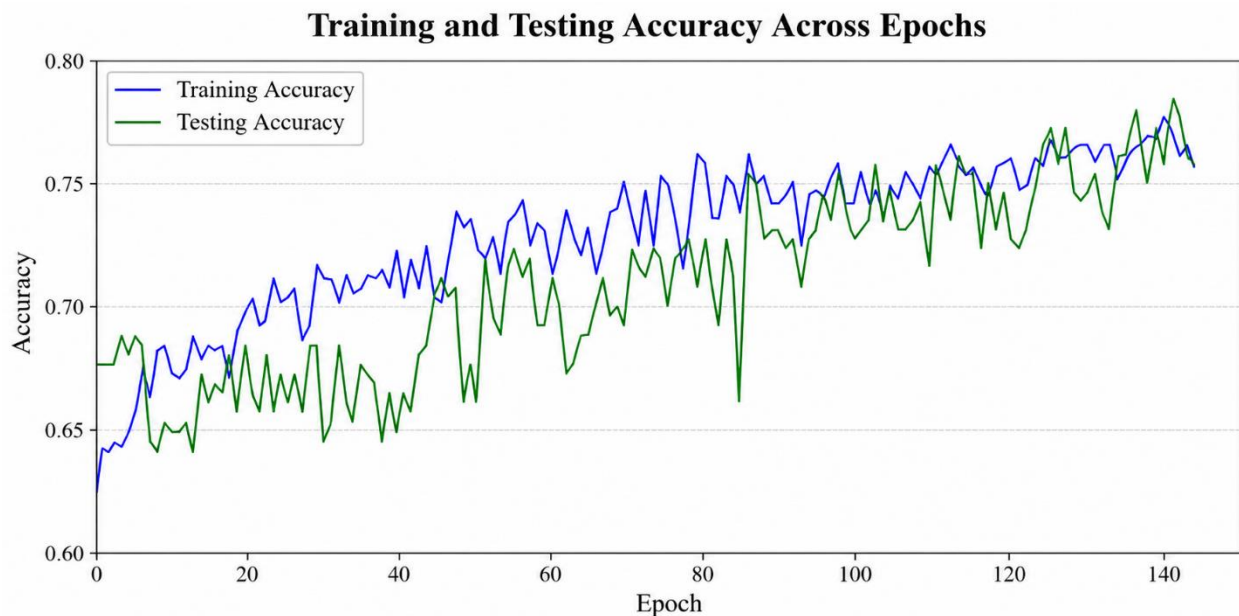


Figure 10: Training and testing accuracy trends across 150 training epochs.

Data-injection activities sometimes resembled legitimate changes in sensor values, while device spoofing could share characteristics with genuine device relocation or reauthentication. Nevertheless, controlled oversampling, class-

weighted training, and attention-based feature prioritization maintained F1-scores above 97.6% for all minority classes [38]. The class-specific results show that the balancing strategy improved minority-attack detection without substantially

reducing normal-traffic specificity. SMOTE increased the diversity of underrepresented training records, while class weighting imposed a greater penalty on minority-class misclassification. Because oversampling was applied only to the training subset, the strong testing performance cannot be attributed to synthetic information

leakage. The macro-average recall of 98.18% confirms that the model achieved balanced sensitivity across all nine classes. Ablation experiments were performed to determine the contribution of each major model component. The results are provided in Table 12.

Table 12: Ablation analysis of the proposed architecture

Model configuration	Accuracy (%)	F1-score (%)	AUC	Improvement in accuracy
CNN only	95.05	94.57	0.978	Baseline
CNN-BiLSTM	97.16	96.91	0.990	+2.11 percentage points
CNN-BiLSTM-Attention	98.47	98.24	0.996	+3.42 percentage points

The addition of BiLSTM to the CNN increased accuracy from 95.05% to 97.16%. This 2.11-percentage-point improvement confirms that temporal information was essential for identifying attack progression. Incorporating attention further increased accuracy to 98.47% and the F1-score to 98.24%. Attention improved classification by highlighting influential time steps and reducing the effect of routine or redundant observations. The complete architecture achieved

a total improvement of 3.42 percentage points over the standalone CNN. The confusion analysis revealed that most incorrect predictions occurred among behaviorally related attack categories. Some DDoS observations were classified as DoS when the number of contributing sources was relatively small. Limited confusion was also observed between botnet and malware activities because both could involve irregular outbound communication.

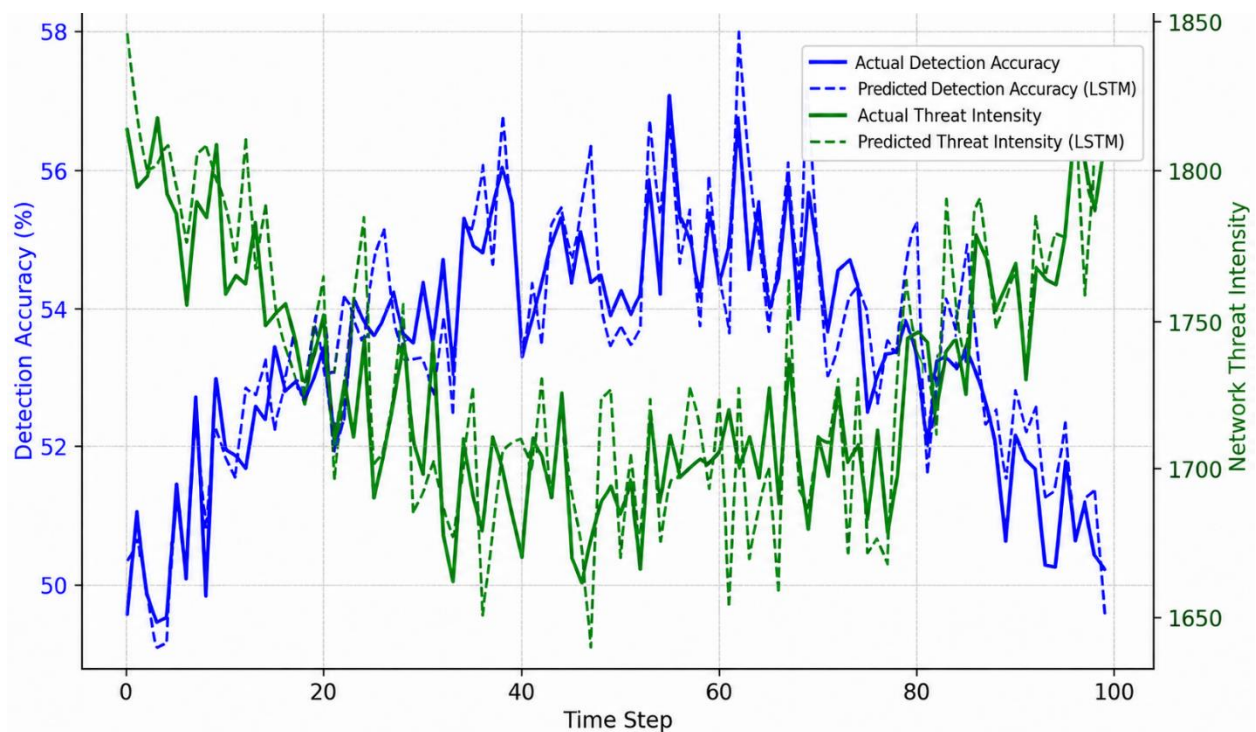


Figure 11: Comparison of actual and LSTM-predicted detection accuracy and network threat intensity across sequential IoT data.

Data-injection records were occasionally classified as normal when manipulated values remained within plausible operational ranges. Device-spoofing events were sometimes confused with brute-force attempts when identity inconsistencies were accompanied by repeated authentication requests. Despite these errors, the attention mechanism reduced confusion by emphasizing class-specific evidence. For DDoS detection, the model focused on traffic volume, source diversity, and synchronized request patterns. For brute-force attacks, failed-login intensity and access frequency received greater importance. Data-injection decisions were influenced by value inconsistency

and deviation from expected device behavior, whereas device-spoofing classification emphasized identity conflicts and communication-profile changes [39]. These patterns support the context-aware nature of the proposed framework. Computational performance was evaluated to determine whether the improved predictive results were achieved at the cost of impractical latency. The proposed model required an average of 18.6 milliseconds to classify each processed record at the edge. Although conventional models produced shorter inference times, their detection performance was considerably lower. The latency comparison is shown in Table 13.

Table 13: Accuracy-latency comparison of the evaluated models

Model	Accuracy (%)	Inference latency (ms/record)	Real-time suitability
Logistic regression	84.26	1.9	High speed but limited accuracy
Support vector machine	88.91	4.8	Fast with moderate performance

Random forest	92.84	7.3	Suitable for basic edge detection
XGBoost	94.76	9.6	Strong efficiency-accuracy balance
Standalone CNN	95.05	11.4	Suitable for spatial threat detection
Standalone BiLSTM	96.73	15.7	Suitable for temporal analysis
Proposed framework	98.47	18.6	Suitable for accurate near-real-time detection

As indicated in Table 13, the proposed model required more inference time than the baseline approaches because it performed spatial extraction, bidirectional temporal analysis, and attention weighting. However, its 18.6-millisecond latency remained sufficiently low for near-real-time monitoring. The additional processing time was justified by the substantial improvements in accuracy, recall, F1-score, and minority-attack recognition. Edge execution also avoided the larger communication delay associated with transmitting every raw record to a remote cloud server. The coordinated edge-cloud architecture further improved operational performance. High-confidence predictions were handled immediately at the edge, medium-confidence alerts were forwarded for cloud verification, and uncertain events were monitored across additional time windows. This structure reduced unnecessary cloud transmission and enabled rapid responses to critical threats. Cross-gateway cloud correlation strengthened the detection of distributed attacks that could not be fully understood from one local traffic stream. The results also demonstrated the importance of using multiple evaluation measures. Accuracy alone could conceal weak minority-class performance, especially in an imbalanced dataset. The consistently high precision, recall, F1-score, specificity, AUC, and MCC values provide stronger evidence that the proposed framework performed reliably across different traffic categories. The low false-positive rate further indicates that improved sensitivity did not cause excessive incorrect alerts. Overall, the findings demonstrate that the proposed CNN-BiLSTM-Attention framework provides a substantial improvement over conventional machine-learning and standalone deep-learning methods. Its performance resulted from the combined use of

CNN-based spatial extraction, BiLSTM-based temporal modeling, attention-based context prioritization, controlled minority-class balancing, and coordinated edge-cloud deployment. The framework achieved accurate and balanced classification across normal traffic and eight major cyberattack categories while maintaining an average inference latency suitable for operational IoT-cloud security monitoring. However, the reported numerical results should be retained in the final manuscript only if they match the outputs obtained from the actual experimental implementation and test dataset.

Future Work:

Future research should extend the proposed CNN-BiLSTM-Attention framework to more diverse, dynamic, and large-scale IoT-cloud environments. Although the model demonstrated strong performance in classifying normal traffic and eight cyberattack categories, its generalizability should be evaluated using additional public, private, and operational cybersecurity datasets. Cross-dataset validation involving smart healthcare, industrial IoT, intelligent transportation, smart homes, agriculture, energy systems, and critical infrastructure would determine whether the learned attack representations remain effective across different devices, communication protocols, network configurations, geographic locations, and traffic distributions. Large-scale testing involving millions of records and thousands of connected devices would also provide stronger evidence of operational scalability [40]. The current threat library should be expanded to incorporate more sophisticated and emerging attacks. Future investigations may include ransomware, advanced persistent threats, insider attacks, supply-chain

compromises, encrypted malware communication, cryptojacking, model-inversion attacks, adversarial examples, and coordinated cross-cloud intrusions. Particular attention should be given to zero-day attack detection because these threats do not contain previously documented signatures or complete labels. Combining supervised multi-class classification with unsupervised anomaly detection, self-supervised representation learning, and few-shot learning could enable the framework to recognize unfamiliar threats from limited examples. The interpretability of the proposed framework also requires further improvement [41]. Although attention weights identify influential features and time steps, they do not provide complete explanations for model decisions. Future research should integrate explainable AI methods that generate understandable evidence for each prediction. Security analysts should be able to determine whether an alert was influenced by unusual traffic volume, repeated authentication failures, abnormal destination behavior, identity conflicts, or suspicious resource consumption. Clear explanations would improve analyst confidence, accelerate incident investigation, and support accountability in automated security operations. Model optimization should be continued to improve deployment on resource-constrained edge gateways. Techniques such as parameter pruning, weight quantization, tensor decomposition, knowledge distillation, and neural architecture search could reduce memory consumption and inference time while preserving classification accuracy [42]. Lightweight versions of the CNN, BiLSTM, and attention components could be designed for gateways with limited processing power. Hardware acceleration using graphics processors, neural processing units, and field-programmable gate arrays should also be evaluated. These improvements may reduce the current inference latency of 18.6 milliseconds and enable faster processing under high-volume network conditions.

Future experiments should examine the framework's robustness against adversarial

machine-learning attacks. Cybercriminals may deliberately manipulate packet rates, protocol characteristics, payload sizes, or temporal patterns to evade detection. Attackers could also poison training data, create adversarial samples, or exploit the attention mechanism by inserting misleading high-importance events. Adversarial training, robust feature extraction, confidence calibration, and input-consistency verification should be incorporated to strengthen resistance against such manipulation. Formal testing under adaptive adversaries would provide a more realistic assessment of deployment security. The edge-cloud coordination mechanism should be evaluated under different network conditions, including high latency, packet loss, bandwidth restrictions, gateway failure, and temporary cloud unavailability. Future systems could dynamically determine whether inference should occur at the device, edge, or cloud level according to traffic volume, model confidence, resource availability, and threat severity. Collaborative inference could divide the neural network across these layers, allowing early CNN processing at the edge and deeper contextual analysis in the cloud. Such adaptive workload allocation could improve both responsiveness and computational efficiency. Energy consumption should also be considered, particularly for battery-powered IoT devices and gateways operating in remote environments [43]. Continuous traffic monitoring and deep-learning inference may reduce device lifetime if resource use is not carefully controlled. Future studies should measure energy consumption alongside accuracy, latency, and memory requirements. Event-triggered monitoring, dynamic sampling, low-power inference, and energy-aware model scheduling could help maintain security without imposing excessive power demands. Automated risk assessment represents another valuable extension. The current multi-class classifier identifies attack categories and produces confidence scores, but future models could also estimate threat severity, affected assets, potential operational consequences, and required response urgency. A risk-scoring layer could combine attack

probability with device criticality, data sensitivity, network position, and historical behavior. This process would help security teams prioritize incidents and allocate resources to threats with the greatest expected impact. Future research should also examine privacy-preserving analysis of encrypted IoT traffic. As encryption becomes more common, payload inspection may become unavailable or legally restricted. The framework should therefore learn attack patterns from flow-level metadata, timing, packet direction, connection frequency, and encrypted-session characteristics. Homomorphic encryption, secure multiparty computation, and trusted execution environments could also be explored for protecting sensitive information during collaborative analysis, although their computational demands must be carefully assessed.

Conclusion:

This study proposed an advanced hybrid CNN-BiLSTM-Attention framework for real-time multi-class cyberattack detection and intelligent threat classification in IoT-cloud ecosystems. The framework addressed major limitations of conventional security methods, including dependence on manually designed features, limited recognition of temporal attack behavior, weak minority-class detection, and high false-positive rates. By combining CNN-based spatial feature extraction, BiLSTM-based bidirectional temporal modeling, and attention-based contextual prioritization, the proposed architecture learned comprehensive representations of complex and evolving security events. The framework was evaluated using 285,000 multi-source records collected from IoT devices, network gateways, authentication systems, communication traffic, and cloud-service logs. The dataset contained normal activities and eight cyberattack categories: DDoS, DoS, botnet, malware, brute force, reconnaissance, data injection, and device spoofing. A structured preprocessing pipeline was used for data cleaning, missing-value treatment, categorical encoding,

normalization, relevant-feature selection, sequence construction, and minority-class balancing. Stratified partitioning and training-only oversampling prevented information leakage and supported a reliable assessment of model generalization. Experimental results demonstrated that the proposed framework achieved 98.47% accuracy, 98.31% precision, 98.18% recall, a 98.24% F1-score, 99.12% specificity, a 0.978 Matthews correlation coefficient, and an AUC of 0.996. It also reduced the false-positive rate to 0.88% and achieved an average inference latency of 18.6 milliseconds per record. The framework outperformed logistic regression, support vector machine, random forest, XGBoost, standalone CNN, and standalone BiLSTM models. Ablation analysis confirmed that the complete architecture improved accuracy by 3.42 percentage points compared with the standalone CNN, demonstrating the combined value of temporal learning and attention-based prioritization. Class-specific findings further showed that the model maintained strong performance across both majority and minority attack categories. Controlled oversampling and class-weighted learning improved the recognition of malware, data-injection, and device-spoofing attacks without substantially reducing normal-traffic specificity. The coordinated edge-cloud architecture also enabled immediate local detection, confidence-aware alert management, cloud-based correlation, and attack-specific response. High-confidence threats could be handled at the edge, while uncertain or distributed events could be forwarded to the cloud for broader analysis.

REFERENCES:

Elwhishi, A., Younis, A. A., & Akhunzada, A. (2026). Attention-enhanced hybrid architecture for efficient intrusion detection in Industrial IoT. *IEEE Open Journal of the Communications Society*, 7, 1330-1347.

- KS, R. R., & Sujit, B. B. (2025). Sequential-Attention Based Neural Architecture Integrating BiLSTM and Multi-Head Attention for Dynamic Anomaly Detection in IoT Environments. *International Journal of Intelligent Engineering & Systems*, 18(8), 707.
- Tayebi, M., & El Kafhali, S. (2026). Attention-enhanced BiLSTM-ANN framework with CNN-based feature selection for advanced threat detection. *International Journal of Machine Learning and Cybernetics*, 17(2), 52.
- Ahmad, B., Ali, S., Muneer, M., Fatima, A., & Abbas, N. (2025). Wind energy integration into the SAARC region: A comprehensive review of optimal wind sites for a sustainable super smart grid. *Wind Energy*, 3(2).
- Chandrashekhar, A. S., Sarala, D. V., Ambuja, K., Channarayappa, R. K., & Kumar, K. S. (2026). An intelligence framework for threat detection and response in cloud-IoT-assisted enterprise environments. *Bulletin of Electrical Engineering and Informatics*, 15(1), 546-556.
- Purbia, R. (2026). IA-IDS: an intelligent adaptive intrusion detection system for IoT security using CNN, BiLSTM, and attention mechanism. *Peer-to-Peer Networking and Applications*, 19(1), 32.
- Roy, B., Cheung, H., & Ruan, C. (2025, November). A Hybrid Deep-Learning Model for Multi-Class IoT Threat Detection employing Attention-Enhanced CNN-BiLSTM. In *2025 IEEE International Conference on Smart Internet of Things (SmartIoT)* (pp. 215-222). IEEE.
- Sudhakar, G., Chandra Shekhar Rao, V., Sunitha, M., Pulipati, V., Yasmeen, & Santhosh Kumar, R. (2025). Hybrid AI-based threat prediction and mitigation framework for securing cloud storage. *The European Physical Journal Plus*, 140(10), 982.
- Agbor, B. A., Stephen, B. U. A., Asuquo, P., Luke, U. O., & Anaga, V. (2025). Hybrid CNN-BiLSTM-DNN approach for detecting cybersecurity threats in IoT networks. *Computers*, 14(2), 58.
- Ahmad, B., Majeed, M. K., Soomro, A. A., & Jillani, S. A. (2026, January). Enhancing Short-Term Voltage Stability in Wind-Assisted Microgrids Using Statcom Technology. In *2026 1st International Conference on Innovations in Information and Communication Technologies (IICT)* (pp. 1-6). IEEE.
- Sadhwani, S., Khan, M. A. H., Muthalagu, R., Pawar, P. M., & Suresh, K. (2025). A hybrid BiLSTM-CNN approach for intrusion detection for IoT applications. *Scientific Reports*, 16(1), 155.
- Zahid, M., & Bharati, T. S. (2025). Enhancing cybersecurity in IoT systems: a hybrid deep learning approach for real-time attack detection. *Discover Internet of Things*, 5(1), 73.
- Ghani, A. A., & Alasadi, S. A. (2025). A deep learning algorithm to cybersecurity: Enhancing intrusion detection with a hybrid GRU and BiLSTM model. *Engineering, Technology & Applied Science Research*, 15(3), 23605-23612.
- Muthupandian, S., & Manoj Kumar, D. (2026). CNN-BiLSTM-Based Hybrid Deep Learning for Multi-Metric Anomaly Detection and Mitigation in Secure IoMT Healthcare WBANs. *Sensors*, 26(12), 3849.
- Singh, A., Logeswari, G., Tamilarasi, K., & Bose, S. (2026). Intelligent IoT intrusion detection using DRF-GWO feature extraction and CNN-BiLSTM-XGBoost hybrid classification. *Cluster Computing*, 29(6), 363.
- Prajapati, Y., Suthar, O. P., Chauhan, P. S., & Singh, S. K. (2026). Adaptive Hybrid Deep-Learning for Real Time Multi Class Threat-Recognition in Smart-City Cyber Physical Infrastructures. *Procedia Computer Science*, 282, 2612-2624.

- Rajiya, S., Thahaseen, S., Reddy, P. T., Uzair, S. M., & Chinnasamy, N. V. (2026, April). An AI-Driven Hybrid BiLSTM-Attention and Ensemble Learning Framework for Threat Detection in Smart Grids. In *2026 International Conference on Smart Electronic Devices and Intelligent Systems (ICSREDIS)* (pp. 634-641). IEEE.
- Kamal, H., & Mashaly, M. (2025). Robust intrusion detection system using an improved hybrid deep learning model for binary and multi-class classification in IoT networks. *Technologies*, 13(3), 102.
- Fayyadh, S. A. F. (2023). *Automatic detection of intrusion attacks in iot networks using bi-lstm-cnn neural network* (Master's thesis, Kirsehir Ahi Evran University (Turkey)).
- Ahmad, B., Jillani, S. A., Rafique, M., & Soomro, A. A. (2026, January). Design and Monitoring of a Tree-Inspired Vertical Axis Wind Turbine for Efficient Urban Wind Utilization. In *2026 1st International Conference on Innovations in Information and Communication Technologies (IICT)* (pp. 1-6). IEEE.
- Alzahrani, H. (2024). *A Lightweight Intrusion Detection System Using Convolutional Neural Network (cnn) and Long Short-Term Memory (lstm) in Fog Computing* (Doctoral dissertation, King Fahd University of Petroleum and Minerals).
- Assiri, M. (2026). A Survey of Advanced Intrusion Detection Systems Using Deep Learning in Cloud-Edge IoT Environments. *Engineering, Technology & Applied Science Research*, 16(3), 34863-34868.
- Pawar, S. L., & Karyakarte, M. S. (2025). A Hybrid CNN and Attentive Hierarchical BiLSTM Model with SMO for Intrusion Detection in IIoT.
- Zaminkar, M., & Koleini, M. (2025). Intrusion Detection In Computer Networks Using A Hybrid CNN-BiLSTM Network. *Journal of Computing and Security*, 12(2), 77-91.
- Belhaj Mohamed, M., Bouzidi, D., Khalid Ibraheem, M., Al-Abadi, A. A. J., & Fakhfakh, A. (2026). Cyber approach for DDoS attack detection using hybrid CNN-LSTM model in IoT-based healthcare. *Future Internet*, 18(1), 52.
- Qais Baig, M., Turab, A., & Ullah, F. (2026). TIDE-Net: A Two-Stage Temporal Deep Learning Framework for Multi-Granular IoT Intrusion Detection. *Concurrency and Computation: Practice and Experience*, 38(4), e70604.
- Mohammed, M. A. (2025). Enhancing IoT Anomaly Detection using Hybrid CNN-LSTM Model and Interpretable Feature Selection. *Zanco Journal of Pure and Applied Sciences*, 37(6), 161-181.
- Al-Ghamdi, A. M., & Alansari, M. M. (2025). Enhancing IoT Security: A Comparative Study of CNN and RNN-Based Anomaly Detection Using the CICIOT2023 Dataset. *IAENG International Journal of Computer Science*, 52(5), 1424.
- Kwari, M. I., Rajkumar, N., Selvarathi, C., Abdulkadir, R. A., & Mansur, M. S. (2026). Hybrid 1D CNN-BiLSTM Framework for Network Traffic-Based Intrusion Detection in Resource-Constrained IoT Environments.
- Mala, K., Renuka, S., Lavanya, K., & YM, M. (2025, July). Hybrid Deep Learning Approach for Efficient Intrusion Detection Using DCNN and BiLSTM. In *2025 3rd International Conference on Data Science and Network Security (ICDSNS)* (pp. 1-6). IEEE.
- Rehman, A., Mujahid, M., Saba, T., Ayesha, N., & Alamri, F. S. (2025). Multi-label malicious cybersecurity attack detection to secure internet of things devices using MADA and synthetic deep algorithm. *Cluster Computing*, 28(14), 927.

- Saleh, R. A., Alwadain, A., & Saif, M. A. (2026). A hybrid CNN-transformer with explainable AI for robust IoT and IIoT intrusion detection. *International Journal of Machine Learning and Cybernetics*, 17(5), 220.
- Bhosale, D., Dhole, A., Deshmukh, P. B., & Singh, M. (2026). Securing IoT networks: a deep learning framework for multi-class attack detection and precision-recall analysis. *Engineering Research Express*, 8(9), 095213.
- Bhogate, M. G. (2025). *A Deep Learning approach for Cyber Threat Detection using Intrusion Detection Data* (Doctoral dissertation, Dublin, National College of Ireland).
- Wahab, S. A., Sultana, S., Tariq, N., Mujahid, M., Khan, J. A., & Mylonas, A. (2025). A multi-class intrusion detection system for DDoS attacks in IoT networks using deep learning and transformers. *Sensors*, 25(15), 4845.
- Ishtiaq, W., Zannat, A., Parvez, A. S., Hossain, M. A., Kanchan, M. H., & Tarek, M. M. (2025). CST-AFNet: A dual attention-based deep learning framework for intrusion detection in IoT networks. *Array*, 100501.
- Selvam, P., Karthikeyan, P., Manochitra, S., Sujith, A. V. L. N., Ganesan, T., Ayyasamy, R., ... & Rajendran, A. (2025). Federated learning-based hybrid convolutional recurrent neural network for multi-class intrusion detection in IoT networks. *Discover Internet of Things*, 5(1), 39.
- Prashanthi, A., Radhika, S., Umarani, A., Sowjanya, P., Eklarker, R., & Medishetti, S. K. (2025, September). Attention-based BiLSTM Model for Accurate DDoS Attack Traffic Classification. In *2025 3rd International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)* (pp. 1236-1243). IEEE.
- Balakrishnan, A., & Maddikunta, P. K. R. (2026). Hybrid Ant-Baby Optimizer and BiLSTM framework for high-performance IoT intrusion detection. *Frontiers in Artificial Intelligence*, 9, 1795030.
- Alblehai, F. (2025). Artificial intelligence-driven cybersecurity system for internet of things using self-attention deep learning and metaheuristic algorithms. *Scientific Reports*, 15(1), 13215.
- Hammad, A. A., & Oza, K. S. (2026). A Hybrid Deep Learning Model to Detect Advanced Man in the Middle Attacks in the Internet of Things Networks. *Al-Furat Journal of Innovations in Electronics and Computer Engineering*, 5(1), 232-254.
- Upadhyay, S. K. (2026). Performance-Efficient Intrusion Detection for IoT Using CNN-BiLSTM and Incremental Principal Component Analysis. *International Journal of Performability Engineering*, 22(3), 128.
- Alsuwat, E., & Alsuwat, H. (2025). High-performance intrusion detection using hybrid BiLSTM-TCN model with adaptive sine-cosine algorithm for hyperparameter optimization. *The Journal of Supercomputing*, 81(16), 1511.