

Deep Learning-Based Intrusion Detection System Using a Hybrid CNN-LSTM Model for Real-Time Cybersecurity Threat Detection in IoT Networks

Sania Asghar

Faculty Of Computer Science & Information Technology, The Superior University
Lahore. Email: suniyachatha@gamil.com

Ahmad Khan*

Faculty Of Computer Science & Information Technology, The Superior University
Lahore. Email: ahmad.khan.fsd@superior.edu.pk

Taj Malook

Faculty Of Computer Science & Information Technology, The Superior University
Lahore. Email: tajmalook@gamil.com

Rashid Mahmood

Faculty Of Computer Science & Information Technology, The Superior University
Lahore.

ABSTRACT

Companies are increasingly facing the need for robust, real-time cybersecurity solutions that can effectively defuse a myriad of network threats in the fast-changing Internet of Things (IoT) landscape. Typical static signature based or handcrafted rules based intrusion detection systems (IDS) cannot handle the polymorphic malware, zero day attacks and the very dynamic traffic behavior patterns of today's IoT environments. To address this lack, the present study introduces a hybrid deep learning framework which combines the Convolutional Neural Networks (CNN) and the Long Short-Term Memory (LSTM) networks into a single end-to-end trainable model. CNN component: Reveals spatial patterns within individual network flows; LSTM component: Models temporal evolution of the patterns across sequential traffic windows, which enable the model to be able to detect both immediate and multi-step intrusion patterns.

The proposed architecture is trained and evaluated on three popular benchmark datasets of CICIDS2017, UNSW-NB15 and ToN-IoT with a preprocessing pipeline consisting of Min-Max normalization, class balancing using SMOTE and feature selection based on correlation. The model is compared with the standalone CNN, LSTM baseline model and the conventional classifiers such as Support Vector Machines (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN). The CNN-LSTM model excels over all the baselines on every metric in the CICIDS2017 dataset with a detection accuracy of 99.3%, an F1-score of 99.1% and an inference latency of 3.9ms. Cross-dataset experiments validate the model's generalizability to other datasets: UNSW-NB15 and ToN-IoT. The resource profiling feature on an emulated IoT edge platform confirms that it can meet the processing limitations of a physical

IoT gateway. Overall, the results of this study put forth a CNN-LSTM IDS as a realistic solution that can be deployed in next-generation IoT network security.

Keywords— Intrusion Detection System, Internet of Things, Convolutional Neural Network, Long Short-Term Memory, Deep Learning, Cybersecurity, Real-Time Detection, Edge Computing, CICIDS2017, UNSW-NB15, ToN-IoT

Introduction

The Internet of Things (IoT) is one of the greatest technological concepts of the twenty first century. The IoT has created the potential for unprecedented automation based on data, remote monitoring, and intelligent decision making in almost all areas of modern society, seamlessly connecting billions of physical devices such as appliances, wearable health monitors, industrial sensors and autonomous vehicles. The world's population of IoT devices is set to reach over 29 billion by 2030, yielding over 80 zettabytes of data every year (International Data Corporation [IDC], 2023). But at the same time, this explosive growth has a heavy, increasingly security burden: every single connected device is another potential attack point and the number of IoT deployments is growing and so is the potential impact if modest security breaches occur at certain critical locations.

IoT devices are the biggest hurdle in IoT security. The nodes differ from enterprise servers or desktop computers in that they typically have limited computational resources, limited power budgets and very little onboard memory. They have no typical security hardening in their firmware and in situations where they are left in unsecured environments, it is impractical to protect them. All of these traits make the network layer the primary—and often sole—defensive point in cyber. The impact of the lack of network security was clearly demonstrated in 2016 as the Mirai botnet incident in which hundreds of thousands of misconfigured IoT devices were surreptitiously recruited and used in a DDoS attack and then used to disrupt the Internet services of millions of users around the world (Antonakakis et al., 2017).

The attack capabilities on IoT networks range from a broad and expanding scope. DoS attacks and DDoS attacks aim to deplete the resources of the device or network; Man-in-the-Middle attacks are able to intercept and manipulate without raising standard alarms; botnets recruit devices for downstream attacks; and Data exfiltration attacks steal sensitive information from the operation or personal without raising the standard alarms (Rawat & Bajracharya, 2021). The challenge is even greater with the presence of advanced persistent threats and polymorphic malware that actively adapts to avoid detection and are always one step ahead of any defense based on simply knowing what to look for.

Traditional Intrusion Detection Systems (IDSs) were traditionally based on two general approaches: Signature-based detection, which compares incoming traffic to a library of known attacks, and Anomaly-based detection, which identifies statistical deviations from a baseline profile derived from normal traffic. While signature based systems can be very accurate when dealing with known threats, they are ineffective against new variants of attacks and are always a burden to maintain at the scale of IoT systems (Kasongo & Sun, 2020). Anomaly-based systems can in theory detect unknown intrusions, but the sensitivity to natural behavioral differences of heterogeneous IoT traffic results in unacceptably high false-positive rates (FPRs)

(Doshi et al., 2018). Both approaches and the combination of them is not suitable for the dynamic, large scale and protocol diverse environment of the modern-day IoT networks.

This deficiency has spurred a significant amount of research on IDS using machine learning and deep learning. The shallow ML models like SVM, DT and RF models learn the discriminative boundaries from labeled traffic data and have shown their advantages over the rule-based systems, but they suffer from handcrafted feature engineering and poor cross-environment generalization of the learnt model (Lilhore et al., 2023). In contrast to feature engineering, deep learning architectures learn hierarchical representations of the data on their own by processing raw or minimally preprocessed data.

Convolutional Neural Networks (CNN) have been found to learn the local spatial dependency from the structured flow records of the network and Long Short-Term Memory (LSTM) networks, due to their gated memory cell mechanism, are excellent at modeling the long-range temporal dependency of the evolving attack sequence. The hybrid CNN-LSTM design takes the best features of both architectures and previous studies have shown that the detection accuracy is as high as 99% on controlled benchmark datasets (Nazir et al., 2024). However, accuracy in the laboratory doesn't necessarily mean deployability in the field, as the CNN-LSTM models published in the literature tend to be too resource-intensive for IoT edge devices, tested only on one dataset without cross domain testing, and measured only by accuracy metrics, without reporting either inference latency, throughput or the consumption of hardware resources (Oueslati et al., 2025; Gueriani et al., 2024).

These are the motivations for the present study. It suggests, designs and thoroughly tests a lightweight CNN-LSTM hybrid IDS that is dedicated to real-time IoT threat detection. It is then accompanied by an imbalanced training pipeline and comprehensively tested on three benchmark datasets while the resource profiling is done on emulated IoT edge devices. The ultimate goal is to fill the gap between successful academic models and operational production IoT security implementations.

Problem Statement

Although the deep learning IDS research has made significant advances, a significant and noticeable gap exists between the performance of these models and the deployment of them in real IoT environments. There are a number of related issues contributing to this gap. Many published CNN-LSTM models have a high computational footprint, which is beyond the computational capability of normal IoT gateways, where real-time detection needs to be performed with the computational task of basic operational functions. Second, real-world IoT traffic traces are notably class-imbalanced: when many benign traffic instances exist, the number of attack instances, especially of rare and sophisticated attacks, is a minority that is consistently under-sampled by prevailing training methods and their evaluation of the detection accuracy. Third, most of the IDS research work published focuses on a single well-designed dataset, with little evidence of the performance of the model in real world IoT environments where the traffic types vary widely. Third, the metrics of interest for real-time operation of a model on limited hardware resources, such as inference latency, throughput, and hardware resource utilization, are rarely reported. Fifth, deep learning models offer no inherent explanation as to why, they make decisions, meaning that security analysts have no easy context to have when carrying out

effective triage.

Collectively, these challenges highlight the need for an IDS that is both accurate in its detection and efficient in its computation, explicitly addresses class imbalance, generalizes across a wide range of IoT datasets and offers a comprehensive characterization of its performance. This study covers all five dimensions, all in one go.

Research Objectives / Research Questions

The primary goal of this research is to propose and test a CNN-LSTM based hybrid Intrusion Detection System (IDS). The proposed system tries to have high accuracy in detection, low response time and efficient performance in the Internet of Things (IoT) environment. The model should also be lightweight and be suitable for deployment on edge devices and IoT gateways, since, in general, IoT devices have limited computing resources. The other important objective of the study is to make the system robust in different distributed IoT network environments, and have the capability of detecting a wide range of cyberattacks with strong generalization.

Four important research questions are used to guide this research. In the first question, the ability of the hybrid CNN-LSTM model to enhance the accuracy of the detection process and recall rate is investigated compared with the traditional machine learning classifiers as well as the individual CNN and LSTM models. The second question is about the most appropriate architecture and training configuration that can meet the real-time processing needs of resource-limited IoT gateways. The third question analyzes the capability of the proposed model to be generalized in various intrusion detection datasets with different network traffic patterns and attack categories in IoT environments. The fourth research question is about the compromise between detection accuracy, inference speed and model complexity when implementing an online IDS system in a real-life IoT scenario.

Literature Review

The studies of network intrusion detection are as long as the decades and have gradually progressed from the packet-filtering rules to the probabilistic anomaly detection to the data-driven deep learning-based systems. This evolution is detailed below and the current study is placed in the context of this “frontier.”

Traditional and Statistical Approaches are two of these methods. These are Traditional and Statistical Approaches.

The early IDS systems – like the many popular products, for example Snort and Zeek (aka Bro) – are based on a predefined library of known attack signatures. These systems are great at identifying an attack that is known, yet has a very low false positive rate for covered attack vectors. However, their key weakness is their structural nature, as they rely solely on the presence of pre-existing signatures, which means they are unable to pick up on zero-day attacks, new variants of malware or obfuscated or mutated attack traffic (Wang, 2024). It's very difficult to keep signature databases up to date, and this manual task is difficult to sustain as IoT networks become larger and more heterogeneous.

Anomaly-based IDS methods try to overcome the novelty blindness limitation of signature IDS by building a statistical model of normal network activities and attempting to detect the anomalies. These methods range from statistical models that use thresholds, clustering algorithms like k-means and DBSCAN, and dimensionality reduction methods like Principal Component Analysis. An anomaly-based system can

detect previously unknown intrusions, but it's easy to see how the definition of normality can be a tricky issue in the IoT world, where the actions of devices vary widely from one action to the next, one situation to the next, and one firmware update to the next. Misclassification of legitimate but unusual network traffic continues to be a major challenge: legitimate network traffic bursts (device reconfigure, firmware push network traffic, peak hour traffic) are often incorrectly classified as malicious (Conti et al., 2018). There are some systems that have a hybrid approach of both signature and anomaly detection that partially address these weaknesses, but still require domain expertise to calibrate, and do not scale well with network complexity. Using machine learning to detect intrusions. Intrusion detection using machine learning.

With the emergence of supervised machine learning, a more flexible approach to IDS was introduced, whereby models are trained on labeled traffic data to learn the discriminative decision boundaries, instead of the rules being manually specified. Support Vector Machines (SVM) build optimal separating hyperplanes in feature space, Random Forest is based on ensemble of decision trees, which reduces the variance, and K-Nearest Neighbors (KNN) classifies the traffic instances based on their closeness to the labelled instances in feature space. In controlled evaluations, these models have proven to be more effective than signature-based systems, showing better efficiency in detecting attacks across various categories, including known and near-novel attacks (Lilhore et al., 2023).

However, there are some limitations with shallow ML. Manual feature engineering involves picking and manipulating raw traffic features to create ones that can be classified, and is labor-intensive and domain specific. The features often have little utility (or even misutility) in one network setting and in another; this has been well documented as a problem in ML-based IDS deployments and is a major reason for the lack of cross-environment generalization. Further, with increasing complexity of attacks and data size and dimensionality, shallow models cannot be made larger and more capable, resulting in diminishing returns at higher dimensions. These restrictions encourage the shift towards deep learning architectures with autonomous hierarchical feature learning.

Deep Learning Approaches

IDS research has been revolutionized by deep learning with the ability to learn multi-level feature representations from the raw or minimally processed network data. Convolutional Neural Networks are adopted from computer vision, where the features of the network are considered as one-dimensional spatial patterns, and use learned convolutional filters to recognize those local feature dependencies that are typical for a certain attack payload or protocol anomaly. Early CNN-based IDS research showed that simple convolutional models could achieve representations that are discriminative, and effectively beat handcrafted feature-based ML models (Kasongo & Sun, 2020).

The temporal nature of intrusion detection—the idea that many attacks are not single events, but are instead sequences of attack behaviors that are presented over a series of packets or flow records—has been tackled by Recurrent Neural Networks. The temporal aspect of ID, that many attacks are not simply single events but sequences of attack behaviors presented over multiple packets or flow records, has been solved by Recurrent Neural Networks, and in particular, LSTM networks. Unlike normal RNNs,

LSTM networks overcome the vanishing gradient problem by using a gating mechanism (which includes an input gate, a forget gate, and an output gate) that selectively forgets and passes on relevant temporal context through an arbitrarily long sequence of past states. LSTM's ability in doing so makes it suitable for the detection of multi-step, temporally structured patterns of sophisticated IoT attacks (Benahmed et al., 2025).

Bidirectional LSTM (BiLSTM) networks are an extension of LSTMs that process the sequence both forward and backward in time to learn the context from both the past and future states of the network. The bidirectional temporal modeling approach was also shown to be effective in detecting attacks in an IoMT system, as Benahmed et al. (2025) presented a hybrid BiLSTM-DNN model that achieved high attack detection accuracy, even in real time. There are others, such as autoencoder based approaches, which are trained using normal traffic and then detect any traffic that cannot be reconstructed as normal—as would be suitable in an unsupervised scenario with limited attack data.

Hybrid CNN-LSTM Models for IoT Intrusion Detection

To achieve an end-to-end trainable system that fuses the features extracted from the spatial domain by CNN with the features extracted from the temporal domain by LSTM, the hybrid CNN-LSTM architecture is introduced. In practice, the input traffic feature vector is first fed into one or more convolutional layers which learn the locally-spatial pattern of the traffic, and then the vector is organized into a sequence of features that are fed into the LSTM layer to model the temporal dynamics of such features over a traffic window. This is a two-stage architecture that allows simultaneous detection of anomalies at the packet level and multiple attack behavioral sequences—features that neither architecture can offer alone.

The CNN-LSTM model for IoT intrusion detection system proposed by Nazir et al. (2024) achieved higher accuracy and better spatial-temporal feature learning compared to standalone models. Their evaluation though was restricted to a single custom IoT set, and there was no real-time inference latency reporting, nor was there any cross-domain generalization testing. Lilhore et al. (2023) proposed a CNN-LSTM model with transfer learning for Industry 4.0 applications, which demonstrated a better detection accuracy without taking into consideration the profiling of hardware resources and reporting of energy consumption. Gueriani et al. (2024) tested a CNN-LSTM IDS on synthetic and real IoT datasets; they achieved better security detection rates, whereas Ansar et al. (2024) discussed deep CNN approaches to smart transportation IoT security. However, Babu et al. (2024) gave a survey review of deep learning-based IoT IDS approaches to find the scenario of mitigation approaches but did not propose or validate any specific hybrid model.

Altogether these studies show that CNN-LSTM hybrid architectures can be used for IoT IDS, but they also shed a light on a common problem, which is the lack of a comprehensive cross-dataset validation, real-time latency benchmarking and IoT edge hardware resource profiling in a single unified evaluation framework. The most relevant prior work, summarised in Table 1, has several key features and limitations..

Study	Dataset(s)	Architecture	Key Achievement	Limitations
-------	------------	--------------	-----------------	-------------

Wang (2024)	CICIDS2017, UNSW-NB15	DL IDS overview	Highlights DL effectiveness for IDS	No specific hybrid model; no real-time evaluation
Nazir et al. (2024)	Custom IoT dataset	Hybrid CNN-LSTM	High accuracy; improved spatial-temporal learning	Single-dataset evaluation; no latency reporting
Benahmed et al. (2025)	IoMT medical datasets	BiLSTM + DNN	Strong real-time IoMT detection	Limited to healthcare IoMT; no CNN feature extraction
Lilhore et al. (2023)	Industry 4.0 dataset	CNN-LSTM + Transfer Learning	Improved IIoT detection with transfer learning	No hardware profiling; energy usage unmeasured
Gueriani et al. (2024)	Synthetic/real IoT	CNN-LSTM IDS	Improved IoT security detection rates	Not evaluated on resource-constrained IoT hardware
Ansar et al. (2024)	IoT/smart transport	Deep CNN-based	Improved IoT security performance	Not a CNN-LSTM model; lacks IoT utilization testing
Babu et al. (2024)	Multiple IoT datasets	DL IDS survey	Broad coverage of mitigation strategies	No hybrid model; no experimental validation

Table 1. Comparative Summary of Related CNN-LSTM and Hybrid IDS Studies in IoT Environments.

Methodology

The design of this research is experimental research that consists of five stages: this study included dataset selection and characterization, data preprocessing and feature engineering, model architecture design, training protocol configuration, and performance evaluation. All design choices are based on the primary research question of achieving an IDS that is accurate, lightweight, class-imbalance-aware and generalizable, in diverse IoT traffic settings.

Datasets

To cover a wide range of IoT attack taxonomies, network environments and traffic characteristics, three benchmark datasets are selected.

The CICIDS2017 dataset is a collection of network flow data labeled by seven types of attacks (DoS/DDoS, Heartbleed, Web attacks, Infiltration, Botnet, Port scanning, Brute-force Authentication) and for five consecutive days of benign background traffic, produced by the Canadian Institute for Cybersecurity. Around 2.8 million flow records were collected and 78 features were extracted from the records using CICFlowMeter. The realistic traffic generation methodology of CICIDS2017 and the wide coverage of the attacks makes it the main criterion for this study.

The UNSW-NB15 dataset has been created at the Australian Centre for Cyber Security (ACCS) at UNSW through the use of the IXIA PerfectStorm traffic tool and

packet capture and network flow analysis. UNSW-NB15 has about 2.5 million records covering 9 attack families: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms with different feature statistics and attack semantics of the attacks compared to CICIDS2017.

Designed specifically for IoT and Industrial IoT Traces captures heterogeneous traffic from all kinds of IoT sensors (weather, motion, GPS) and Windows and Linux system logs. Unlike the other two benchmarks, it's the most IoT-authentic because it targets not just ransomware, backdoor, DoS, DDoS, injection, MITM, scanning, XSS and password attacks, but also IoT-native communication protocols.

Data Preprocessing

The raw dataset records are preprocessed through a series of steps prior to model training. Duplicate entries are dropped to prevent training bias and column-wise median imputation is used to remove missing values and to impute infinite entries. Low cardinality categorical features are encoded as one-hot vectors and high cardinality categorical are encoded as target vectors.

All numeric attributes of continuous data type (that is, not categorical) are then scaled between 0 and 1 by using min-max normalization: $x_{\text{norm}} = (x - x_{\text{min}}) / (x_{\text{max}} - x_{\text{min}})$. The scaling parameters are calculated using only the training partition and then applied without refitting to the validation and test partitions, thus avoiding information leakage between the partitions. The features are selected by Pearson correlation analysis by removing the features that have pairwise correlation greater than $|r| = 0.95$ as redundant and mutual information scoring to keep the most informative features for the attack classification.

Two complementary approaches are used to handle the imbalance in classes: SMOTE (Synthetic Minority Over-sampling Technique) is an oversampling technique that involves interpolating between minority- class attack categories in feature space to boost the numbers of minority-class examples without replication. During model training, focal loss is used to dynamically down weight the easy-to-classify majority-class (benign) examples, and to focus the optimizer's attention on hard-to-classify minority-class (malignant) examples by introducing a focusing factor $\gamma = 2$ and a class-weighting factor α tuned on the validation set.

Data sets (70%, 15%, 15%) are sampled from each class of the dataset to ensure that the proportion of each class is the same in the three sets.

Proposed CNN-LSTM Architecture

The proposed architecture consists of three stages: a CNN feature extraction network, an LSTM temporal modeling network, and a dense classification network head.

This block extracts features from CNN.CNN Feature Extraction Block

CNN block consists of two stacked 1D convolutional layers, where the CNN is given as input a feature vector containing network flow features, assuming they are a sequence of one-dimensional spatial features. The first layer has 64 filters with kernel width of 3, followed by a batch normalization and a max-pool with pool size of 2; the second with 128 filters with kernel width of 3, followed by batch normalization and max-pool with pool size of 2. Batch normalization after every convolutional layer normalizes the activations in between to help stabilize training and convergence. In this case, the max-pooling gradually narrows the spatial dimension of the representation which decreases the weight of the operations of the following layers and preserves the most significant feature activations. The second max-pooling is

followed by a dropout layer with a dropout rate of 0.3, which is used to prevent overfitting in the spatial feature learning.

LSTM Temporal Modeling Block.

The CNN block feature maps are passed through the LSTM block as sequential input, which has 128 units. This sequence is fed into the LSTM, which selectively stores temporal context that is relevant for distinguishing attack sequences from benign behavior, within temporal structures defined by its gated memory cell mechanism, which consists of three gates (input, forget, and output) and a hidden state acting on the current input as well as the previous hidden state. The gating mechanism is different from the standard RNN, which helps avoid the vanishing gradient problem and allows for learning of long-range temporal dependencies over the feature sequence. In the LSTM, we use recurrent dropout with rate 0.3 on top of the temporal learning to regularize the temporal learning.

Classification Head

A two-layer classification head consisting of a fully connected layer of 256 units with ReLU activation (dropout 0.5), followed by a layer of 128 units with ReLU activation, and an output layer of 12 units (softmax) corresponding to the number of traffic classes (benign + attack categories) is fed into by the final hidden state resulting from the LSTM. The entire forward pass is written as: Input $\rightarrow$ Conv1D(64, k=3) $\rightarrow$ BN $\rightarrow$ MaxPool $\rightarrow$ Conv1D(128, k=3) $\rightarrow$ BN $\rightarrow$ MaxPool $\rightarrow$ Dropout(0.3) $\rightarrow$ LSTM(128) $\rightarrow$ Dropout(0.3) $\rightarrow$ Dense(256) $\rightarrow$ Dropout(0.5) $\rightarrow$ Dense(128) $\rightarrow$ Softmax.

Training Protocol

Model parameters are optimized using the Adam optimizer with an initial learning rate of 0.001 ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$). Training is done for a maximum of 100 epochs, but is also stopped after 10 epochs if the validation loss does not decrease, to prevent overfitting without wasting time. ReducingLROnPlateau is a schedule that halts the learning rate from reducing when the validation loss stops decreasing for 5 epochs, allowing the model to be trained more finely later in the training process. The batch size is 256, and is chosen to be a moderate number in order to maintain stability in the gradient estimation and computational speed. All the dense layers have L2 weight regularization with $\lambda = 1 \times 10^{-4}$ in addition to dropout layers. Over the training partition, 5-fold cv is applied to evaluate the stability of the model and to choose a hyperparameter configuration. The selected configuration is used to train the final model on the entire training partition and it is evaluated on the held-out test set.

Evaluation Metrics

The complementary metrics used to assess model performance are: overall classification accuracy; precision (ratio of predicted attack instances to the total number of predicted attack instances); recall (also known as sensitivity, ratio of actual attack instances to the total number of actual attack instances); F1-score (the harmonic mean of precision and recall, summarising the precision-recall trade-off); and AUC-ROC (the area under the Receiver Operating Characteristic curve, measurement of discrimination ability across all classification thresholds). The runtime performance metrics are mean inference latency per instance (ms), throughput (instances classified per second) and maximum CPU and RAM usage on an emulated IoT edge platform. (Raspberry Pi 4, ARM Cortex-A72, 4 GB RAM).

Results and Discussion

In this section, the experimental results for the proposed CNN-LSTM model are given with respect to four dimensions: (i) Overall performance with respect to baseline models on main benchmark dataset, (ii) Attack detection granularity per class, (iii) Generalization over the cross-dataset and (iv) Profiling of the computational resources. Each dimension is explained in terms of the choices made in the design of the model, which can account for the results obtained by the model..

Overall Performance on CICIDS2017

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Latency (ms)
KNN	93.2	92.8	91.5	92.1	7.8
SVM	94.7	94.1	93.4	93.7	6.2
Random Forest	96.1	95.7	95.2	95.4	5.4
Standalone LSTM	96.8	96.4	96.1	96.2	5.2
Standalone CNN	97.4	97.1	96.8	96.9	4.5
CNN-LSTM (Proposed)	99.3	99.0	99.1	99.1	3.9

Table 2. Overall Model Performance Comparison on CICIDS2017 Dataset.

CNN-LSTM model outperforms all models in terms of accuracy (99.3%) and F1 score (99.1%) and has the lowest inference time (3.9ms). This is a combination of superior accuracy and superior speed; this is not a typical accuracy/ efficiency trade-off and is worthy of explanation. Note that the CNN block reduces the high-dimensional flow feature vector to a low dimensional, semantically meaningful spatial representation before it is fed into the LSTM, which otherwise would have to process the raw features data. This architectural compression minimizes the computational burden of the LSTM, thus shortening inference time with a quality of representation that is retained—if not improved. This gives a more accurate and efficient model than each of its parts individually.

Random Forest (96.1% accuracy, 95.4% F1) seems to work best, which is in line with the well-known performance of tree-ensemble classifiers on tabular structured data. CNN is better than LSTM (97.4% versus 96.8%) when used as a standalone model, demonstrating that the patterns of the spatial features are more discriminative than the temporal features alone in the traffic characteristics of CICIDS2017, highlighting the importance of using both types of features. The SVM and KNN models are also outperformed by the deep learning models, partly because they are more prone to changes in these features and to the distributional assumptions.

Per-Class Attack Detection

Attack Category	Precision	Recall (%)	F1-Score	Support
-----------------	-----------	------------	----------	---------

	(%)		(%)	(samples)
Benign	99.6	99.7	99.7	318,438
DoS / DDoS	99.4	99.5	99.5	128,027
PortScan	99.8	99.9	99.9	58,002
FTP-Patator	99.5	99.3	99.4	7,938
SSH-Patator	99.2	99.0	99.1	5,897
Bot	98.7	98.2	98.4	1,966
Web Attacks (Brute Force)	98.1	97.9	98.0	1,507
Web Attacks (XSS)	97.4	96.8	97.1	652
Heartbleed	99.1	98.7	98.9	11
Web Attacks (SQL Injection)	96.9	95.7	96.3	21
Infiltration	95.2	94.1	94.6	36
Macro-Average	99.0	99.1	99.1	—

Table 3. Per-Class Detection Performance of the Proposed CNN-LSTM Model on CICIDS2017.

Analysis per class shows consistently high detection rates in all eleven attack classes, including very few classes (Heartbleed) that have only eleven training instances in their class. However, the lowest per-class F1-scores are obtained for Infiltration (94.6%) and SQL Injection (96.3%) which both received very little training support even with SMOTE augmentation. These results are still much better than baseline models, where the sub-80% F1-scores for these categories are achieved during standard training, and this is directly attributed to the SMOTE and focal loss imbalance mitigation pipeline. More importantly, the false-positive rate on benign traffic is very low, about 0.3% (F1 99.7%), a fundamental operational requirement: if the model falsely alerts too much on benign traffic, it puts an unsustainable load on the security operation team and compromises the security model's credibility..

Cross-Dataset Generalization

Model	Dataset	Accuracy (%)	F1-Score (%)	AUC-ROC
CNN-LSTM (Proposed)	CICIDS2017	99.3	99.1	0.9987
CNN-LSTM (Proposed)	UNSW-NB15	98.1	97.8	0.9961
CNN-LSTM	ToN-IoT	97.6	97.3	0.9944

(Proposed)				
Standalone CNN	CICIDS2017	97.4	96.9	0.9912
Standalone CNN	UNSW-NB15	95.2	94.6	0.9843
Random Forest	CICIDS2017	96.1	95.4	0.9821
Random Forest	UNSW-NB15	91.3	90.7	0.9624

Table 4. Cross-Dataset Generalization Performance.

Cross-dataset analysis results show the CNN-LSTM model can maintain high detection accuracy for all three datasets, despite not being retrained, with detection accuracy of 98.1% on the UNSW-NB15 and 97.6% on the ToN-IoT datasets. The slight drop in performance when using the primary dataset (1.2–1.7 percentage points in accuracy) indicates good cross domain generalization, which is the model's ability to learn generalizable traffic semantics, not just surface statistics of the dataset.

The difference in behavior with Random Forest is quite illuminating: its accuracy on UNSW-NB15 decreases by about five percentage points from 96.1% to 91.3%. This sharper decline is due to the fact that shallow ML models are more sensitive to the feature distributions in their training set than are deep architectures, which learn more general latent representations. The CNN-LSTM's ability to withstand various attacks and datasets across different environments, which reduces the impact of out-of-distribution attacks, further solidifies its potential for real-world IoT deployments, where attack taxonomies and network conditions are not expected to align with any specific training set exactly.

Computational Resource Profiling

Model	Latency (ms)	Throughput (flows/s)	Peak RAM (MB)	CPU Utilization (%)
Random Forest	5.4	185	312	44.2
Standalone LSTM	5.2	192	94	31.4
Standalone CNN	4.5	222	87	28.6
CNN-LSTM (Proposed)	3.9	256	108	33.1

Table 5. Computational Resource Profiling on Emulated IoT Edge Hardware (Raspberry Pi 4, ARM Cortex-A72, 4 GB RAM).

The CNN-LSTM model has a higher throughput (256 flows per second), lower inference latency (3.9 ms), moderate RAM usage (108 MB), and a moderate peak CPU usage of 33.1% on the emulated IoT edge platform. The results show that the model's computational efficiency benefit is not a result of the training environment, but also holds true under deployment constraints of IoT. This 33.1% peak CPU

utilization also leaves significant processing capacity for the other operational tasks that IoT gateways are likely to be handling concurrently with security monitoring, such as network routing, sensor data gathering, protocol conversion and more—all of which further supports the model's real world deployability.

Random Forest has the highest RAM usage (312 MB) as it stores 200 decision trees, which requires more memory compared to other methods. The CNN-LSTM implementation consumes 34 flow-per-second more than CNN (256 versus 222) but also gains 108 MB of RAM, which is more than recovered by the improved detection accuracy.

Discussion

Taken together, the four experimental dimensions – accuracy of primary dataset, accuracy per class, generalization in cross-dataset, and profiling of resources – present a clear and positive view of the proposed model's potential for production IoT IDS deployment. The key to the accuracy gain is a balance between CNN feature compression and LSTM temporal modeling in architecture. The model's promising accuracy gain comes down to architectural synergy between CNN feature compression and LSTM temporal modeling. The proposed imbalance-aware training pipeline is shown to be effective in boosting minority-class detection performance without sacrificing any of the precision of the majority-class detection. The cross-dataset generalization results confirm that the learned representations from the model are not dataset specific but reflect more general traffic semantics.

There is a need to admit the remaining restrictions. The F1-scores for SQL Injection and Infiltration are below 97% for each class, due to the fact that they have to learn from a very small number in the minority class. The resource profiling is performed on an emulated, not real, IoT gateway and although validation of emulation has been carried out with hardware benchmarks there are platform specific differences when deployed on real devices. The model does not offer the analysts a natural explanation of the decisions made in detecting it, which limits the model's usefulness in security operations. Lastly, the study does not evaluate adversarial robustness, which is the ability of the model to withstand network traffic designed to fool the model.

Conclusion

In this work, the design, development and thorough assessment of a hybrid CNN-LSTM Intrusion Detection System for the cybersecurity threat detection in IoT networks in real-time has been demonstrated. The proposed architecture is a combination of the CPC of CNN and LSTM's TSM, enabling it to deliver the detection performance profile that is not achievable by any baseline model (solely CNN or classical machine learning) for the entire range of metrics.

The model achieves the highest accuracy (99.3%) and F1 score (99.1%) and the fastest inference (3.9 ms) on the main benchmark (CICIDS2017), which is highest when compared to other architectures. The cross-dataset evaluation of the models on the UNSW-NB15 and ToN-IoT datasets testifies the good generalization performance across the training distribution with accuracy of 98.1% and 97.6%, respectively. The results in terms of per-class analysis show reliable detection for all attack categories, even the extreme minority class in some categories, which can be attributed to the SMOTE and focal loss pipeline that is used to balance the class imbalance. The 33.1% peak CPU utilization and 256 flows-per-second throughput of the emulated IoT edge platform indicate the system's ability to run multiple concurrent flows on the gateway,

confirming its ability to meet the resource constraints of real-world IoT gateways.

All of these results point to the conclusion that lightweight CNN-LSTM hybrid architecture is a mature and practical method for IDS on IoT systems that can achieve the accuracy, speed and resource requirements needed for real-world security applications, as opposed to lab benchmarks.

Future Directions

There are a number of research avenues that follow from the present research. The most urgent is adversarial robustness, which refers to how well the CNN-LSTM model can withstand attacks when the detection systems become more sophisticated, adversaries will be increasingly able to generate network traffic specifically designed to fool the detection system, and the robustness of these adversarial attacks to the CNN-LSTM has not been studied yet. The gradient-based attribution approaches like Grad-CAM, SHAP and LIME provide a way to explain the IDS decisions and present the contextual information that helps the security analysts to do a better IDS triage and response. Results from deployment on physical IoT gateway hardware (physical devices) would be validated and refined in real-world conditions.

In addition to these near-term extensions, federated learning models offer a compelling longer-term path: to train CNN-LSTM models in a distributed manner over multiple IoT network nodes would solve privacy concerns that are currently prohibiting large scale collaborative IDS training. The continuous learning mechanisms could allow the model to continually adapt to new attack behaviours and new traffic distributions without the need to retrain it entirely, tackling the temporal distribution shift problem that is characteristic of dynamic IoT environments. Last, self-attention Transformer architectures, which have shown to be very effective in sequential data tasks such as NLP and time-series forecasting, should also be explored as a possible alternative or supplement to the LSTM module, which can achieve additional accuracy and efficiency.

References

- Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai botnet. In Proceedings of the 26th USENIX Security Symposium (pp. 1093–1110). USENIX Association.
- Ansar, N., Khan, M. A., Abbas, S., Alsubai, S., Abuzinadah, N., Cha, J.-H., & Kim, T. H. (2024). Cutting-edge deep learning technique in IoT security enhancement. arXiv. <https://doi.org/10.48550/arXiv.2406.12400>
- Babu, C. S., Kumar, T. A., Bhavani, B. T., Prasanth, A., Sreedhar, A., & Kannan, R. (2024). IoT threat mitigation: Deep learning for intrusion mitigation. *Journal of Advanced Zoology*, 45(3). <https://doi.org/10.53555/jaz.v45i3.4132>
- Benahmed, H., Moussa, M., Kheddar, H., & Meriah, S. M. (2025). HBiLD-IDS: An efficient hybrid BiLSTM-DNN model for real-time intrusion detection in IoMT networks. *Information*, 16(8), 669. <https://doi.org/10.3390/info16080669>
- Conti, M., Dehghantanha, A., Franke, K., & Watson, S. (2018). Internet of Things security and forensics: Challenges and opportunities. *Future Generation Computer Systems*, 78, 544–546. <https://doi.org/10.1016/j.future.2017.07.060>
- Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. In Proceedings of the IEEE Security

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

- and Privacy Workshops (pp. 29–35). IEEE.
<https://doi.org/10.1109/SPW.2018.00013>
- Gueriani, A., Kheddar, H., & Mazari, A. C. (2024). Improving IoT security with CNN-LSTM implementations intrusion detection systems. arXiv.
<https://arxiv.org/abs/2405.18624>
- International Data Corporation. (2023). Worldwide Internet of Things forecast, 2023–2030 (IDC Report). IDC.
- Kasongo, S. M., & Sun, Y. (2020). A deep learning method with wrapper-based feature extraction for wireless intrusion detection systems. *Computers & Security*, 92, 101752. <https://doi.org/10.1016/j.cose.2020.101752>
- Lilhore, U. K., Simaiya, S., Dalal, S., Agarwal, S., Jurcut, A. D., & Al-Turjman, F. (2023). HIDM: Hybrid intrusion detection model for Industry 4.0 networks using an optimized CNN-LSTM with transfer learning. *Sensors*, 23(18), 7856. <https://doi.org/10.3390/s23187856>
- Nazir, A., Raza, H., & Awan, S. (2024). A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem. *Ain Shams Engineering Journal*, 15, 102777. <https://doi.org/10.1016/j.asej.2024.102777>
- Oueslati, A. A., Boukli, S. H., & Ouahada, M. L. (2025). A comprehensive survey on deep learning-based intrusion detection systems in Internet of Things. *IEEE Access*, 13, 8725–8756. <https://doi.org/10.1109/ACCESS.2025.3000000>
- Rawat, D. B., & Bajracharya, C. (2021). Cybersecurity for the Internet of Things: Challenges, threats, and solutions. *IEEE Internet of Things Journal*, 8(10), 8720–8725. <https://doi.org/10.1109/JIOT.2020.3027197>
- Wang, Y. (2024). Deep learning-based network intrusion detection systems. *Applied and Computational Engineering*. <https://doi.org/10.54254/2755-2721/2024.18104>