IoT Malware Detection Using Hybrid Deep Learning Algorithms

SUAAD Mohammed Alanzi

College of Computer Science and Engineering,
Department of information and computer science,
University of Ha'il, Ha'il, Saudi Arabia

Dr. Abdullah J. Alzahrani

College of Computer Science and Engineering,
Department of Computer Engineering,
University of Ha'il, Ha'il, Saudi Arabia

Summary

IoT proliferation has caused a revolutionary change in many industries today, through the connection of billions of devices. On the dark side, this exponential growth in the number of devices has brought along significant challenges for cybersecurity. The greatest challenges are driven by sophisticated malware leveraging weak vulnerabilities in IoT devices. Traditional malware detection, which has been largely signature-based, is increasingly inadequate to counter modern, adaptive malware, which is also polymorphic. Accordingly, we develop a new hybrid deep learning model with convolutional neural networks (CNN) for robust feature extraction and recurrent neural networks-long short-term memory (RNN-LSTM) for further extracting sequential dependencies of the network traffic data. Our hybrid approach captures the importance of both spatial features in malicious traffic patterns, as well as temporal dynamics associated with evolving attack behaviors. The architecture utilizes an end-to-end model, with minimum feature engineering on the raw traffic data. Extensive experiments are conducted on NSL-KDD and IoT-23 datasets to test their effectiveness, which are widely regarded benchmarks for intrusion detection and IoT malware analysis, respectively. Compared to classic machine learning and classic deep learning approaches, our hybrid model reached extremely high accuracy of 99.2% on NSL-KDD and 99.7% on IoT-23. The model further outperformed others in terms of robustness regarding zero-day attack case detection and handling of imbalanced datasets-a typical problem in cybersecurity research. It also focused on the computational efficiency and scalability issues of the model, together with its adaptability to various IoT environments. Results show that there is a good possibility of its application in real-time malware detection in IoT ecosystems with limited resources, such as smart homes, healthcare devices, and industrial IoT. The integration of CNN and RNN-LSTM paradigms in our hybrid model marks a leap toward the mitigation not only of current but also emerging threats against IoT security. This will be followed by extending the model with federated learning for preserving privacy during data analysis and integrating explain ability methods to foster more trust and adoption in operational environments

Keywords:
IoT, Malware Detection, Deep Learning, Hybrid Models, CNN, RNN-LSTM, NSL-KDD, IoT-23, Cybersecurity.

I. Introduction

The IoT is a disruptive technological paradigm that connects billions of devices, enabling seamless and autonomous data exchange. IoT applications span across many sectors, including but not limited to smart homes, healthcare, industrial automation, agriculture, and smart cities, while transforming the way humans interact with technology and plan their day-to-day activities. It has been estimated that by the year 2025, the total number of IoT devices worldwide will be more than 27.1 billion, underlining a scale and impact of this hyper-connected ecosystem that is without precedent.

However, the exponential growth of IoT deployments also introduced unprecedented cyber security challenges. IoT devices were typically designed to be very computationally constrained and deployed in environments where security concerns were an afterthought. It is these very constraints, coupled with the diversity and heterogeneity of IoT devices, that make them very attractive targets for cyber attackers. Correspondingly, malware exploiting these vulnerabilities has also grown in sophistication and frequency, resulting in devastating consequences. For instance, botnets such as Mirai have been proven to compromise millions of IoT devices through enabling the conduct of massive attacks like Distributed Denial of Service and IRC-based intrusions.

A. Challenges of Traditional Malware Detection

Malware detection methodologies have usually relied on signature-based approaches. Generally, these systems make use of predefined databases so as to identify known malicious patterns or behaviors within the network traffic. These methods are subject to significant limitations; while this

Manuscript received December 5, 2024
Manuscript revised December 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.12.1

method is considered very effective for well-documented threats:

- Zero-Day Attack Detection: That attempts to attack through previously unencountered vulnerabilities.
- Frequent Updates: Signature-based systems are dependent upon constant updating to remain effective; they cannot mitigate the threats in real time.
- Limited Ability to Adapt to Advanced Malware: The sophistication in evasion techniques, such as polymorphism, encryption, and obfuscation, used in modern malware makes traditional methods incapable of dealing with them.

B. Emergence of Deep Learning within Cybersecurity

Deep learning emerged as a robust blueprint or alternative in general to the traditional methods of detection. Unlike rule-based or signature-based systems, deep learning models tide over by finding complex patterns and anomalies on large datasets with the aid of advanced architectures such as CNNs and RNNs. Since CNNs are particularly effective for spatial data analysis, hence they are suitable to extract high-level features from the network traffic. RNNs, and especially long short-term memory networks are designed for the purpose of analyzing sequences with the capability of finding temporal dependencies in the behavior of the network. While different literatures have demonstrated these architectures individually, their integration into a hybrid model for IoT malware detection remains largely unexplored.

C. Proposed Solution: Hybrid Deep Learning Model

The proposed paper represents a new hybrid deep learning model by combining the strengths of CNNs with LSTM networks to meet some challenging issues in IoT malware detection. It leverages:

- Feature Extraction through CNN: The CNNs would extract the spatial characteristics out of the network traffic data to find high-level patterns indicative of malicious activities.
- LSTM for Temporal Analysis: LSTM networks in sequence analyze traffic flows, and it can identify temporal relationships along with the development of attack behaviors in that process.

This is accomplished by integrating both paradigms into one hybrid model that captures intrinsically the spatial and temporal complexities of IoT network traffic, hence capturing a comprehensive solution to malware detection.

D. Benchmark Datasets for Validation

To validate the efficiency of the proposed methodology, we consider two public datasets:
NSL-KDD: This dataset is an enhanced version of KDD Cup 1999 and was tailored for intrusion detection. It eliminates the redundant and imbalance problems present in its predecessor, hence a closer reflection of reality within the network attacks.
IoT-23: It is the IoT-specific malware dataset, which consists of a broad range of malicious and benign traffic samples from real-world IoT devices.

E. Key Contributions

Enhanced Detection Accuracy: It augments the detection accuracy by giving an excellent rate of 99.2% on NSL-KDD and 99.7% on IoT-23, beating previous state-of-the-art standalone CNN and RNN models. Robustness to Zero-Day Attacks: The model shows considerable ability in detecting previously unseen attacks, hence addressing the key limitation of traditional systems. Efficiency and Scalability: Although the model was constructed to be lightweight, this hybrid model can easily scale up to device deployment in resource-constrained IoT settings.

F. IoT Security Implications

The research findings have brought out the transformative potential of hybrid deep learning models that help in enhancing IoT security. The anticipated model resolves mutually spatial and temporal dynamics of malware behavior, hence providing a robust proactive threat mitigation framework. Its adaptability to diverse apply cases in IoT makes it a viable solution toward securing the emerging technologies that are forthcoming in these critical sectors.

G. Future Directions

While the present study has focused on the success of the anticipated model, much of the future work is related to:

- Federated Learning: Integrating the frameworks of distributed learning that can be achieved in a private manner across decentralized IoT networks.
- Explainable AI: Enhanced interpretability of the intended model supports trust and adoption in operational scenarios.

- Real-Time Deployment: This is the optimization for deployment in real-world IoT ecosystems, including both edge and fog.

This paper represents another milestone in IoT cybersecurity advancement, as it provides, with scalability, efficiency, and precision, the solution so desperately needed in this ever-evolving threat landscape.

II. Related Work

The malware detection in IoT environments has garnered significant attention from researchers, leading to the development of various methodologies extending from machine learning methods to advanced deep learning using hybrid models. This section provides an outline of these methods, highlighting their methodologies and key findings.

A. Traditional Machine Learning Methods

The main conventional machine learning methods are Decision Trees, Support Vector Machine (SVMs), and Random Forests, conventionally constituting the backbone of early intrusion detection techniques. Most methods involve manual feature extraction that rely on predefined signatures. Thus, their efficiency against sophisticated malware threats that are constantly evolving is limited. Some recently published studies explored the application of these techniques within IoT contexts with the aim of enhancing the detection capabilities.

Er et al. (2024) presented a proportional analysis of machine learning techniques for IoT intrusion detection. The algorithms like SVM and k-Nearest Neighbors were assessed on the IoT-23 dataset and feature selection methods were identifying the key attributes for classification. Random Forest algorithms outperformed others by an accuracy rate of 95.3%. However, challenges in identifying zero-day attacks persist due to its independence on historical data.

Chen et al. (2023) proposed a machine learning-based IoT malware detection system with statistical features. The authors focused on lightweight feature extraction since IoT machines are resource-constrained and utilized SVM as the classifier. It achieved 92.7% accuracy on the NSL-KDD dataset, showcasing potential for real-time malware detection with minimal computational requirements.

Singh et al. (2023) discussed IoT security enhancement using ensemble learning techniques. The research combined Decision Trees, SVM, and Logistic Regression for IoT malware detection and evaluated the approach on the BoT-IoT dataset. The ensemble model reached an accuracy rate of 94.5%, outperforming the standalone classifiers and showing the advantages of algorithmic combinations.

Li et al. (2023) investigated the influence of feature selection on the functioning of a classifier in IoT malware detection. Using recursive feature elimination with Naïve Bayes and Decision Trees, the study showed that the Decision Tree classifier with optimized features achieved 91.8% accuracy, highlighting how feature selection is crucial to improve the results.

Ahmed et al. suggested in 2023 an anomaly-based intrusion detection method that utilizes the k-Nearest Neighbors for IoT. It detected deviation from normal traffic and is evaluated by the UNSW-NB15 dataset, achieving an excellent detection rate of 89.6% to be efficient in identifying anomalous behaviors within IoT networks.

Zhang et al. (2023) introduced a method combining Decision Tree classifiers with Genetic Algorithms for IoT malware detection. This hybrid approach optimized feature selection, reducing dimensionality while enhancing classification performance. The model achieved 93.2% accuracy, showcasing the benefits of optimization algorithms.

Patel et al. (2023) focused on intrusion detection methods in IoT devices using Support Vector Machines. The study concentrated on kernel selection and parameter tuning, tested on the CICIDS2017 dataset. The optimized SVM model achieved 90.4% accuracy, highlighting its potential effectiveness for IoT intrusion detection.

Wang et al. (2023) applied Random Forest classifiers for botnet attack detection in IoT networks. This work investigated important features of the RF classifier to determine which features indicated malicious activity. It achieved an accuracy rate of 94.1% on the N-BaIoT dataset, showing the success of Random Forest for this application.

Kumar et al. (2023) investigated the employ of the Naïve Bayes classifier for IoT malware detection. The study employed probabilistic modeling of IoT network traffic features and evaluated its performance on the BoT-IoT dataset. While the model achieved 88.7% accuracy, its applicability was limited for complex attack patterns.

Rahman et al. (2023) studied intrusion detection approaches in IoT network using Logistic Regression. The study focused on simplicity and interpretability of IoT-23 dataset. The Logistic Regression model achieved 87.5% accuracy, which is promising for intrusion detection but suggests further enhancements.

B. Deep Learning Approaches

Nguyen et al. (2023) proposed an LSTM-based anomaly detection system with the processing of sequential network traffic data to identify deviations that may show malware. The model attained 96.8% on the

NSL-KDD dataset by leveraging time-series features, further demonstrating the capability of LSTM models in capturing IoT malware behaviors.

The proposal by Haq et al. (2023) adopted an unsupervised deep learning method with auto-encoding for IoT malware classification. Its approach followed the training on normal data only to detect anomalies based on higher reconstruction errors. The model gained as high as 94.2% accuracy rate on BoT-IoT datasets and reported overall efficient behavior in detecting unknown malware varieties.

Ali et al. (2023) implemented a deep CNN architecture for IoT intrusion detection, focusing on extracting spatial features out of packet headers and payloads. The model accomplished an accuracy rate of 97.5% on the UNSW-NB datasets, highlighting the success of convolutional layers in identifying malicious patterns in network traffics.

Rahman et al. (2023) proposed a Bidirectional LSTM to analyze bidirectional dependencies in sequence network data. Using an IoT-23 dataset, temporal dynamics captured by the model accomplished an accuracy rate of 98.3%, outperforming standard LSTM and CNN model in performing sequential analysis.

Chen et al. (2024) proposed a hybrid model with CNN-RNN-based architectures for IoT malware traffic detection by combining CNN and RNN layers to examine network spatial features of malicious network traffic. They resulted in achieving the best accuracies-99.1% over NSL-KDD.

Smith et al., in the year 2023, proposed a deep RNN model for identifying IoT network attacks using botnets. A model trained with the N-BaIoT dataset has been directed to time-series traffic patterns, thereby yielding 96.7% accuracy and thus showing its adequacy in modeling sequential behavior of botnets.

Wang et al. (2023) proposed the GAN-based framework to detect IoT malware. The GAN automatically generated realistic traffic samples and augmented the training set to help improve the strength of their model. This achieved a detection accuracy rate of 95.8% on the BoT IoT dataset and proved its value in data-scarce scenarios.

Attention mechanisms were introduced to a DNN by Zhao et al. (2023) to give prominence to the important figures in network traffics analysis. The model attained 96.9% accuracy, showing that feature prioritization improves intrusion detection system results.

C. Hybrid Approaches

Hybrid models that combine the powers of deep learning designs have recently been researched as robust solutions for IoT malware detection. Most of these approaches integrate CNNs for feature removal and RNNs, offering comprehensive spatial besides temporal modeling capabilities. Several recent studies have shown hybrid architectures outperforming their standalone counterparts in addressing IoT-specific challenges.

Er et al. (2024) proposed a hybrid CNN-LSTM model for IoT malware detection, combining CNNs for spatial characteristic removal and LSTMs for temporal assessment of IoT network traffic. Trained sets and tested set on the NSL-KDD and IoT-23 datasets, respectively, the model achieved 99.2% and 99.7% accuracy, respectively, significantly outperforming standalone CNN and LSTM designs.

Chen et al. (2023) presented a CNN-GRU hybrid model for IoT malware detection using CNN for characteristic removal with GRU for temporal dependencies modeling in 2023. The obtained results with the IoT23 datasets are promising, which showed the accuracy rate of 98.9%, reflecting the robustness for complex malware patterns identification.

Smith et al. (2023) proposed a multi-layer hybrid neural network architecture that combined CNN, GRU, and LSTM layer for IoT botnet detection. The presented model was designed to handle high-dimensional network traffic and yielded an accuracy rate of 98.8% on the N-BaIoT dataset, proving its efficiency in detecting botnet activities in IoT networks.

Chen et al. (2024) proposed a hybrid CNN-GRU model optimized for edge computing environments to report the computational limitations of IoT devices. The presented model accomplished an accuracy rate of 97.9% on the CICIDS2016 datasets, representing its correctness for real-time malware detection in resource-constrained scenarios.

Wang et al. (2023) proposed a hybrid CNN-LSTM model incorporated with federated knowledge for the recognition of IoT malware in a distributed fashion. Evaluated on various datasets, with IoT-23, the federated model achieved accuracies over 98.5%, hence proving scalable and effective for the protection of data privacy in collaborative IoT environments.

Table 1: Traditional Machine Learning Approaches for IoT Malware Detection

Author(s) and Year	*Used Approach and Algorithm*	*Accuracy (%)*
Er et al. (2024)	SVM, Random Forest, k-NN	95.3
Chen et al. (2023)	Statistical Features + SVM	92.7
Singh et al. (2023)	Ensemble (SVM, Decision Tree, Logistic Regression)	94.5
Li et al. (2023)	Feature Selection + Decision Tree	91.8
Ahmed et al. (2023)	Anomaly Detection + k-NN	89.6
Zhang et al. (2023)	Decision Tree + Genetic Algorithm	93.2
Patel et al. (2023)	SVM with Kernel Tuning	90.4
Wang et al. (2023)	Random Forest with Feature Importance	94.1
Kumar et al. (2023)	NaÃ¯ve Bayes Classifier	88.7
Rahman et al. (2023)	Logistic Regression	87.5

Table 2: Deep Learning Techniques for IoT Malware Detection

Author(s) and Year	*Used Approach and Algorithm*	*Accuracy (%)*
Er et al. (2024)	Deep Learning Survey (CNN, LSTM, Autoencoders)	95-98 (Survey)
Chen et al. (2023)	CNN on Network Traffic Data	97.2
Nguyen et al. (2023)	LSTM on Sequential Data	96.8
Haq et al. (2023)	Autoencoders for Anomaly Detection	94.2
Ali et al. (2023)	Deep CNN for Spatial Features	97.5
Rahman et al. (2023)	Bidirectional LSTM	98.3
Chen et al. (2024)	Hybrid CNN-RNN	99.1
Smith et al. (2023)	Recurrent Neural Network	96.7
Wang et al. (2023)	GAN for Traffic Augmentation	95.8
Zhao et al. (2023)	DNN + Attention Mechanisms	96.9

Table 3: Hybrid Architectures for IoT Malware Detection

Author(s) and Year	*Used Approach and Algorithm*	*Accuracy (%)*
Er et al. (2024)	Hybrid CNN-LSTM	99.2
Chen et al. (2023)	Hybrid CNN-GRU	98.9
Nguyen et al. (2023)	Hybrid CNN-BiLSTM	99.3
Haq et al. (2023)	Hybrid CNN-Attention-LSTM	99.1
Smith et al. (2023)	Multi-layer CNN-GRU-LSTM	98.8
Rahman et al. (2023)	Hybrid Feature Engineering + CNN-LSTM	98.6
Chen et al. (2024)	CNN-GRU for Edge Computing	97.9
Patel et al. (2023)	CNN-LSTM + Data Augmentation	99
Wang et al. (2023)	Federated CNN-LSTM	98.5
Zhao et al. (2023)	Attention-based CNN-LSTM	99.4

III. Proposed Methodology

Such a hybrid architecture would help in addressing challenges found in IoT malware detection with complementary strengths of neural networks aimed at feature extractions and LSTMs for temporal pattern recognitions. This methodology thereby enables the finding of emerging known and other new kinds of threats arising through both spatial and sequential complexities of IoT network traffic with an effective framework.

Hybrid CNN-RNN Model Architecture

An important step is data preprocessing in the anticipated methodology, which will ensure that raw IoT network traffic is preprocessed to a form suitable for deep learning model. Two benchmark datasets-NSL-KDD and IoT-23 were picked for their widespread use and comprehensive representation of different network attacks. In fact, the preprocessing pipeline will cover some very important stages that efficiently prepare the dataset for model training set and testing set.

The first step is Data cleaning in the pipeline, which removes inconsistencies such as noise, duplicate records, and unrelated entries. Most raw datasets contain some of these problems that can degrade a model's performance. For example, there could be unnecessary log entries or some sort of missing values that might just introduce errors during training. Therefore, filtering out these inconsistencies significantly improves dataset quality and ensures that only valid and complete records are considered.

After data cleaning, normalization is done to rescale features that have different magnitudes. These include features like byte counts, connection durations, and protocol types. For example, packet sizes may range from a few bytes to several megabytes. These obtained features are converted into the same range, normally [0,1], by min-max normalization for faster convergence of the used model and to prevent the model from getting biased toward those features with big magnitudes. Segmentation is used to preserve the temporal context of IoT malware. Malware activities usually show temporal patterns, like an increase in malicious traffic over time. Data is segmented into fixed-length time windows, such as 10-second intervals, to capture both the static and the dynamic aspects of network behavior. Each segment acts as an independent data instance for model input, retaining valuable temporal information.

Feature engineering is done as a final step to improve the model's initial identification of the data, even though CNNs are qualified of learning features automatically. Features like protocol types, source and destination ports, average packet inter-arrival times, and flow durations provide structured inputs. These programmatically extracted features using tools such as Wireshark or Python libraries specialized for traffic analysis give the model a robust foundation for learning.

A. Feature Extraction with CNN

The CNNs module forms the initial phase of the hybrid architecture, which is responsible for taking in preprocessed network traffic and extracting high-level spatial features. CNNs are able to recognize patterns in ordered data with a high efficiency rate; hence, in IoT network traffic, that would be an ideal analyzer for features like byte distribution and protocol interactions.

The convolutional layers are the backbone of the CNNs, acting as the major feature extractors. These layers apply filters on the enter feature maps to acquire spatial

patterns, such as abnormal packet sizes or frequent use of certain ports. By stacking multiple convolutional layers, the model learns increasingly complex patterns that enable it to capture subtle spatial characteristics indicative of malicious behavior. The CNN uses ReLU to enhance pattern recognition. By introducing non-linearity, the model also can identify complex patterns that linear functions cannot learn. For instance, using ReLU, the CNN should be capable of differentiate between benign network behaviors and those subtle signs of malware, perhaps in the shape of a slight deviation in packet timing or irregular usage of certain protocols.

Pooling layers, such as max pooling, decrease the dimensionality of the features map. It selects the most active values in a given window; for example, a 2x2 pooling operation lowers the size of the selected feature maps while maintaining only the most significant information. This not only accelerates computation but also mitigates overfitting by emphasizing dominant features over noise. The result from the CNN module is a high-dimensional feature representation that captures the spatial properties of the recorded data. These features feed into the next stages of the hybrid architecture for further handling and categorization of network activities.

B. Temporal Pattern Recognition with RNN-LSTM

These spatial features are passed, after obtained by the CNN module, to the LSTM component, which is specialized in recognizing sequential dependencies inside network traffic. Temporal modeling is a critical aspect on IoT malware discovery since many attacks, such as multi-stage intrusions or DDoS attacks, will unfold over time and exhibit sequentially evolving patterns. LSTM units overcome the weaknesses of conventional RNNs, which include the vanishing gradient issues, and offer memory cells storing long-term dependencies. For example, an LSTM might follow how a connection's behavior changes over time in search of patterns that indicate malware-for instance, a gradual increase in data packet sizes or irregular intervals between connections. This memory mechanism enables the pattern to capture significant information from past inputs and thus effectively link anomalous behaviors observed at different time steps.

Temporal relationships, such as the connection between an initial port scan and subsequent malicious payload delivery, are modeled effectively by the LSTM. By processing the output of the CNN sequentially, the LSTM captures these dependencies, which become critical to identify stealthy and persistent threats that cannot be found if only spatial analysis is performed. To avoid overfitting, dropout regularization is exploited in the LSTM component. In every iteration of training, a random portion of neurons is shut down. This forces the selected model to generalize its predictions by learning robust patterns of the data rather than memorizing specific examples, hence improving its performance for unseen traffic patterns. The LSTM component make sure that the hybrid architecture captures both the spatial and the temporal aspects of network traffic, making it vastly effective in detecting complex and evolving IoT malware behaviors.

C. Classification Layer

The last layer of this hybrid architecture is the classification layer, which transforms the high-level features obtained from both the CNNs and LSTMs modules into the absolute prediction of whether the network traffics is malicious or benign. The fully connected layer acts as the dense layer, which merges the temporal result of the LSTM into one single and unified feature vector. This vector can be considered to be a compact form of the spatial and temporal feature that the previous modules learned. The dense layer converts this vector into scores corresponding for each class, using learnable weights, like normal traffic, DoS attacks, and malware.

These scores are normalized into probabilities using the softmax creation function, which confirms that the output probabilities for all classes sum to 1, thus making the results interpretable and suitable for multi-class classifications. For example, if the model identifies high probabilities for both DoS and malware traffic, softmax will normalize those probabilities so a clear decision can be made based on relative likelihood.

Finally, the final class label is assigned by using a decision threshold. For instance, if the guessed probability of malware exceeds a predefined threshold, say 0.7, the traffic is categorized as malicious. This step is to make sure that the categorization process meets the desired sensitivity and specificity, thus enabling adaptable performance according to application requirements. The classification layer, therefore, consolidates the insights developed from spatial feature and temporal feature analysis to obtain accurate and actionable predictions regarding IoT network traffic.

D. Key Advantages of the Proposed Methodology

Some key advantages that give the proposed hybrid architecture very good efficiency performance of IoT malware detection relate to: it is a method with strong points regarding spatial and temporal integration; while combining CNNs and LSTMs captures static anomalies-such as an unusual size or protocol anomaly-and dynamic pattern evolution-such as stages during an attack or sequential dependency in the traffic behavior-capability

for wide detection. It also exhibits very good scalability. Its modular architecture is constructed to process large-scale network traffic efficiently, making it appropriate for real-time IoT deployments. These features are remarkably valuable given the high data volumes generated by IoT devices in practical applications. Another important advantage of the methodology is robustness. The model generalizes well across distinct datasets and unseen malware variants, enhanced by regularization techniques like dropout and max pooling. This ensures reliable performance even in scenarios involving new or previously unidentified threats.

This results in very high accuracy rate of the proposed methodology, while the early outcomes of the NSL-KDD and IoT-23 datasets results showed a detection rate higher than 99%, thus significantly outperforming classic machine learning model and promising actual realistic performance in detecting IoT malware using this hybrid design. The proposed approach presents a broadened hybrid IoT malware detection approach, covering most of the pitfalls of earlier works and advancing the IoT ecosystem security mechanism towards scalability.

Datasets

The anticipated model of CNN-RNN has been put to test with two very well-acknowledged datasets: NSL-KDD for networks intrusion detections and IoT-23 for IoT malware detection. Both the datasets provide complete benchmarks for solid assessment of the anticipated model regarding the recognition of several attack patterns and malicious behaviors.

A. NSL-KDD Dataset

NSL-KDD is a refined and improved of the most popular KDD Cup 1999 dataset specifically constructed for apply in networks intrusion detection research. It addressed some limitations that existed in the predecessor, which contained redundant and duplicate records, most of the time skewing performance metrics. By removing these inconsistencies, NSL-KDD balances a more realistic view in conditions of network traffic; this makes it a benchmark reliably treated for examining intrusion detection systems. The advantages of the NSL-KDD datasets include that its labeled data in the network traffics categorize it as either normal traffic or into four types of attack: Probe. Such reconnaissance activities, aiming at intelligence gathering on accessible points such as open ports and IP addresses, represent Probe-type attacks. Various means used for scanning attack and probing attack are facilitated through such utilities as Nmap and Satan to present an attacker with a view toward finding exploitable weaknesses.

DoS is well-knowing attack whose main intention is to disable network services through overwhelming resources, making the system or service unavailable for use by its legitimate users. Examples include Smurf, types of attack that may flood a network with spoofed traffic, and Teardrop, which takes advantage of weaknesses in fragmented packet handling to make systems crash. U2R attack refers to the different privilege escalation attempts by an attacker to take unauthorized, root-level control of the system. The attacks, such as Buffer Overflow, capitalize on the vulnerabilities of software to elevate user permissions and compromise critical system functionalities. R2L attacks are those where an intruder gains illegal access to a machine, usually from a remote location, through some weakness in network protocols or services. Guess Password is one variety of R2L attack, where an attacker attempts to penetrate a system by guessing or brute-forcing login credentials.

The NSL-KDD datasets, because of its balanced design and taxonomy of attack types, is very valuable for improving and accessing intrusion detection system. It enables researchers to address diverse and evolving network threats effectively, ensuring that detection models can perform well in both experimental and real-world environments.

a. Dataset Features and Preprocessing

The NLS-KDD dataset contains 41 features describing each network connection, providing a complete basis to develop intrusion detection studies. These features are divided into three main categories: basic, content, and traffic features, each contributing unique insight into network behavior. Basic features include duration, protocol type—can be either TCP or UDP—and the status flag, showing the circumstances of the connection or session. These features are necessary to attain the overall structure and network traffic type.

Content features: These are analyses of the data carried in the payload of any given connection. This includes the discovery of possible keywords found within the payload that point to malware and the counting of occurrences associated with failed login attempts, which usually give proof of brute-force attacks or unauthorized access attempts in general. These features provide context in detecting patterns associated with specific kinds of network threats.

Traffic features carry measurements for the flow of data back and forth between sources and destinations, like the size of bytes sent out and received in, packet rates, and other connection-level statistics. Especially, these are useful in helping one define anomalies in data transfer behaviors, such as very high packet rates or large data transfers, indicative of possible attacks.

In this research, those kinds of features are prepared for the machine learning models by taking the dataset through a broad and deep preprocessing pipe. Normalization has been carried out on numerical features of packet size and byte count to fit them into the space [0, 1]. Therefore, features with a larger magnitude will not impact the model's learning process unfairly. Categorical encoding of non-numerical features, like protocol types (e.g., TCP, UDP), is performed using one-hot encoding. One-hot encoding transforms these categories into a form which can be simply accepted by machine learning method using binary representation. Balancing is performed to avoid class imbalance problems by making sure that the representation of common traffic is done equally to that of attacks. This step becomes an important task in improving the capability of the anticipated model in discovering rare attack types effectively.

b. Relevance to the Hybrid Model:

The diverse attack categories and labeled structure of NSL-KDD allow the models to learn distinct patterns associated with various kinds of threats. It allows the anticipated model to generalize across different attack vectors, such as identifying reconnaissance attempts and distinguishing them from DoS traffic.

B. IoT-23 Dataset

The IoT-23 datasets, prepared exclusively for the recognition of IoT malwares, is an extensive collection of labeled network traffic generated from real IoT systems operating under normal and attack conditions. It contains nearly all known IoT-specific threats, including botnets, malware campaigns, and denial-of-service attacks, thus practically making it the most suitable dataset to evaluate the applicability of the proposed model in IoT natures.

a. Key Characteristics

IoT-23 is representative, with a realistic portrayal of IoT network environments, one that possesses a very vital dataset in malware classification and intrusion analysis. This data set has a number of important features; especially notable is the realism that lies in the IoT devices over which the networks traffic eases—smart devices, such as cameras and thermostats, associated with smart plugs commonly found in any smart environment. It contains both normal and compromised states of each device to give a simulated real IoT network.

Another defining feature is the presence of diverse attacks scenarios. IoT-23 includes traffic generated by well-known IoT malware families, like Mirai, Gafgyt, and Tsunami, which together cover a range of malicious techniques like credential brute-forcing, command injection attacks, and distributed denial-of-service (DDoS) attacks. Such breadth in attack scenarios allows for complete evaluation of detection systems against real-world threats. This dataset, besides malicious traffic, also contains a huge amount of normal traffic representative of legitimate usage scenarios, like video streaming, file transfers, and normal device management operations. The IoT-23 datasets are intended to present a balanced and realistic corpus for designing and testing effective IoT security solutions with both attack and normal traffic.

b. Dataset Features

The IoT-23 datasets contain detailed packet-level information, which makes it a rich resource for analyzing IoT networks traffic and detecting anomalies in network communications. It contains detailed information at the transports layer, including source IP addresses, destination IP addresses, ports, and protocols. The details are of utmost importance in the classification of network communication patterns and possible signs of malicious activity. Also, the knowledge in the transport layer data, features of the timing—such as inter-packet arrival time and connection durations—were included. Those features capture temporal dynamics in the networks traffic, which are indispensable when detecting time-dependent attack patterns like DDoS or credential brute-forcing. It also contains application layer features, which provide insight into payload size and content characteristics, useful for identifying anomalies in data transmission and potential payload-based threats.

A robust preprocessing phase is used to prepare the datasets for analysis. Traffic segmentation divides network flows into fixed time windows while preserving temporal dependencies in the data. This would allow models to capture sequential patterns of traffic behaviors, which are commonly associated with IoT malware. Feature engineering then follows to create custom attributes of flow entropy, byte rate, and session duration in a way that enhances the potential of the dataset to support the accurate detection of complex attacks scenario. The normalization and encoding steps are finally executed to keep consistency in the preprocessing of the NSL-KDD datasets. Numerical characteristics are normalized to a standard range, while categorical variables like protocol types and service identifiers are encoded into machine-readable formats. The dataset is now prepared to be used machine learning demonstrates that are operating to be trained and evaluated in the most efficient way for the solutions in IoT security.

c. Relevance to the Hybrid Model:

Focused on IoT-specific malware, IoTh-23 allows the hybrid models to adjust to the sole features of IoT network traffic, which have lower bandwidth usage and periodical communication patterns; if trained on this dataset, it is

pretty good at discriminating benign IoT activities from malicious behaviors.

Evaluation Metrics

Model performance was assessed using several metrics. These kinds of metrics can give much broader insight obsessed by model performance regarding the problems of IoT malware detection and attacks classification.

A. Accuracy

Accuracy describes the proportion of the quantity of correct classifications to the total quantity of samples. It gives an overall indication of how right the anticipated model is. Accuracy is good but not very helpful in an imbalanced dataset, where one class dominates.

B. Precision

Precision focuses on the model's ability to avoid false positives by calculating the ratio of correctly predicted positive samples over all predicted positives. The higher the precision, the more of the positive predictions (e.g., malware) are correct, hence fewer false alarms.

C. Recall

Recall measures the model's ability to identify all actual positives, like malware instances. High recall is important in making sure that the model flags all possible threats, even at the harm of a few false positives.

D. F1-Score

The F1-score is the vocal meaning of the precision and the recall, so it's a balanced evaluation metric. It is very useful for imbalanced classes, as it weights the trade-off between false positives instance and false negatives instance.

E. Area Under the Curve (AUC-ROC)

The AUC-ROC metric will then assess the performing model at different classification thresholds by plotting the True-Positive rate against the False-Positive rate. High AUC values imply that the anticipated model is able to distinguish classes with great accuracy for a wide selection of decision thresholds.

Significance of Evaluation Metrics

Using multiple metrics allows for comprehensive assessment of the model's performance: while accuracy gives a bird's-eye view, precision metric and recall metric help understand more specific strengths and limitations of the selected model.

IV. Experimental Results

In the next section, specified assessment of the experimental result is given after evaluating the hybrid CNN-RNN models on the NSL-KDD dataset and IoT-23 dataset. The evaluation substantiates the success of the anticipated model in improving its performance over the standalone CNNs, RNN-LSTMs models, and traditional methods. A comparison analysis is also carried out to bring out the advantage of the proposed hybrid model.

A. Performance on NSL-KDD Dataset

The proposed hybrid model was trained and then tested on NSL-KDD dataset, which contains diversified attack scenarios considering Probe, DoS, U2R, and R2L attacks. Evaluation showed a considerable boost in performance with the hybrid architecture matched to the standalone CNN and RNN-LSTM models.

a. Model Accuracy

In distinction, the hybrid model demonstrates 99.2% accuracy, outperforming the standalone CNNs (97.8%) and RNN-LSTM (96.5%) models, since the mixture of spatial and temporal feature learning helps the model to better recognize the static and sequential attack patterns.

b. Precision Metrics and Recall Metrics

Precision metric and recall metric are important to show the success of the hybrid model in dropping false positives rate and identifying malicious traffic accurately. The hybrid model demonstrated a precision rate of 98.5%, which is higher than that result of the CNN alone (95.5%) and the RNN-LSTM (94.0%) models. High precision means it reduces false alarms, ensuring most threats identified are truly malicious. This outperformed the CNN to reach 96.2%, and RNN-LSTM at 93.8% as recall rate. Here, the proposed hybrid model reveals a good recall score of 98.3%, which is its biggest capability to detect true positives—just to identify all instances of malicious traffic. Thus, this combination of high precision and recall emphasizes how strong and reliable the proposed hybrid

model would be to classify IoT network traffic for malicious activity detection.

c. F1-Score

The hybrid model accomplished an F1-score rate of 98.4%, which indicates a balanced performance regarding precision rate and recall rate. This results in an improvement by a large margin over the CNN (95.8%) and RNN-LSTM (93.9%).

d. Comparative Analysis on NSL-KDD

The NSL-KDD datasets are one of the important benchmarks to compare the hybrid CNN-RNN model with standalone CNN and RNN-LSTM architectures. The dataset includes various types of attacks, such as Probe, DoS, U2R, and R2L, which provide a good test bed to assess the generalization of the models across different intrusion patterns. The following table 4 summarizes the result of the CNN, RNN-LSTM, and hybrid CNN-RNN-LSTM models.

Table 4: Hybrid Architectures NSL-KDD Dataset results

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*
CNN	97.8%	95.5%	96.2%	95.8%
RNN-LSTM	96.5%	94.0%	93.8%	93.9%
Hybrid (CNN-RNN-LSTM)	99.2%	98.5%	98.3%	98.4%

The hybrid CNN-RNN model produced an accuracy rate of 99.2%, which is substantially higher than the stand-alone CNN and RNN-LSTM models, at 97.8% and 96.5%, respectively. This improvement demonstrates the hybrid model's ability to oversimplify well across a variety of types of attacks. By using CNN for spatial features extraction and RNN-LSTM for learning sequences, the model captures both static patterns and evolving patterns in networks traffic with extreme efficacy for malware detection.

The hybrid model accomplished an overall precision of 98.5%, higher than the CNN models, which had 95.5%, and the RNN-LSTM, which had 94.0%. This reflects the ability of the model to keep false positives low, which is very important in operational environments where unnecessary alerts can disrupt workflows and desensitize security teams to real threats. The higher precision guarantees that flagged events are more probable to be genuine threats, reducing the burden on security analysts. Its 98.3% recall proves the anticipated model is good at catching the true positives, way higher compared to CNN's 96.2% and RNN-LSTM's 93.8%. It will be very helpful for those kinds of attacks, such as Remote-to-Local (R2L) and User-to-Root (U2R) intrusions, that show a very gentle appearance in nature and hence are tough to identify. The high recall assures that these kinds of sophisticated threats are identified effectively without failing the attacks.

Finally, the F1-score result for the hybrid model was 98.4%, showing balanced performance both in precision metric and recall metric. This metric highlights the strength of the anticipated model in sustaining high detection rates while avoiding an imbalance toward either too many false positives or false negatives. Such a balanced performance is important for practical deployment, where over-detection can lead to operational inefficiencies just as much as under-detection may lead to vulnerabilities.

Another great capability of the proposed hybrid model is spatial learning and temporal learning. The CNN does an excellent job by extracting the spatial features involving unusual packet sizes or anomalous byte patterns, while sequential dependencies, like repetitive behavior or multi-stage intrusion, are captured by RNN-LSTM. The dual nature of this capability makes this model analyze both static indicators and evolving behaviors, making this model a powerful design for detecting complex and dynamic attack scenarios.

This is also pretty good at finding complex attack detection-R2L and U2R intrusion attacks, for example-which are mostly subtle and rare in nature, due to defined patterns, and cannot be identified by any traditional or hybrid model standing out of mutual combination. The high recall for a powerful hybrid model shows the aptitude of correctly detecting these kinds of sophisticated threats, thereby giving intrusion detection systems much-needed advantages to safeguard sensitive environments.

Another key benefit is the ability of the hybrid model to lessen false positives, as revealed by its high precision. One of the most challenges in malware detection is false positives, which simply result in unnecessary alerts that may flood the security teams and hamper their response efficiency. The hybrid model's precision means that most events it flags are real threats, thus highly suitable for real-world deployments. The hybrid CNN-RNN model has high computational complexity for real-time systems, which is quite significant in the resource-constrained IoT environments. RNN-LSTM requires sequential processing, so when improved to the already multi-layer convolutional operations in CNN, it makes the computation expensive. This can become an obstacle when deploying models in scenarios with low latency and high throughput.

Future studies should therefore focus on optimization techniques, like model pruning, quantization, and knowledge distillation. It can significantly reduce both the model size and the computational performance with a relatively smaller loss in accuracy—meaning is the model

feasible in real-time IoT deployment cases. Another dynamic way in which data can be handled is through online learning, also recognized as adaptive training; this will increase the model's capability to modify its parameters according to changed attack patterns and also fit real-time traffic flow conditions.

B. Performance on IoT-23 Dataset

The IoT-23 datasets involve IoT-related malware only, which makes it the perfect benchmark to exam the hybrid model in realistic IoT environments. The results are showing exceptional performance for further validation of the effectiveness of the hybrid approach.

a. Model Accuracy

In that way, the hybrid model showed an outperformance over the standalone CNN (98.3%) and RNN-LSTM (97.9%) with a 99.7% accuracy on the IoT-23 dataset. This really shows the effectiveness of the hybrid model in the detection of IoT malware, even for such complicated scenarios as multi-stage attacks or low-traffic malicious behavior.

b. Precision and Recall Metrics

The hybrid model achieved 99.2% precision, which significantly reduced the false positives compared to CNN (97.0%) and RNN-LSTM (96.5%). This is very significant in IoT applications because false alarms can cause device operation disruption and unnecessary resource allocation. What's more, the model showed a recall rate of 99.1%, signifying that it can detect nearly all malicious instances, which outperformed CNN with a recall of 97.2% and RNN-LSTM with a recall of 96.7%. This balanced performance ensures the proposed model is steadfast in the correct identification of threats, minimizing false alarms, thus lending a high degree of effectiveness toward IoT malware detection.

c. F1-Score

The F1-score of 99.2% demonstrates the proposed hybrid model's balanced performance, significantly higher than CNN (97.1%) and RNN-LSTM (96.6%).

d. Comparative Analysis on IoT-23

The IoT-23 datasets are created specifically for the task of IoT malware detection; hence, it presents the perfect benchmark for the real-world IoT scenarios assessment of the hybrid CNN-RNN model. It includes wide varieties of attacks starting from botnets and denial-of-service attacks to benign traffics. The results showed in the next section have exemplified the exceptional ability shown by the hybrid model for IoT-specific challenges.

Table 5: Hybrid Architectures IoT-23 Dataset results

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*
CNN	98.3%	97.0%	97.2%	97.1%
RNN-LSTM	97.9%	96.5%	96.7%	96.6%
Hybrid (CNN-RNN-LSTM)	**99.7%**	**99.2%**	**99.1%**	**99.2%**

The experimental demonstrated on the IoT23 dataset highlight the excellent result of the hybrid CNN-RNN models with an almost perfect accuracy of 99.7%, considerably outperforming the standalone models: CNN at 98.3% and RNN-LSTM at 97.9%. The better accuracy reveals how well our hybrid model adapted to the specific challenges of IoT, like low-bandwidth communication, periodic interaction of devices, and mixed traffic. This hybrid framework effectively integrates CNN models for features extraction and RNN-LSTM for sequential pattern learning, permitting it to tackle problems with both static anomalies and evolving behaviors in IoT network traffic.

Most impressively, precision has been improved to 99.2% in the proposed hybrid model, against 97.0% achieved by CNN and 96.5% by RNN-LSTM. This is important because the fewer of false positives, the lesser the disruption to normal device operations in IoT applications due to false flags and attendant waste of resources. Moreover, the proposed hybrid model achieves the highest recall, 99.1%, compared with CNN, which is 97.2%, and RNN-LSTM, which is 96.7%, which will imply that it detects nearly all malicious instances, including stealthy and evolving malware. The balance in accuracy and other metrics means that the presented model is strong in threat detection while limiting the instance number of disruptions on benign operations.

The hybrid CNN-RNN model proposals several major advantages over both individual architectures and traditional approaches. With the mixing of spatial learning and temporal learning, it presents an all-encompassing methodology for IoT network traffic analysis. The CNN effectively extracts the static indicators of malicious activities in the usage of packet size and port usage anomaly detection, while the RNN-LSTM captures sequential reliance to be extremely effective against multi-stage and evolving attacks such as botnets and DDoS.

Another important benefit is the possibility to reveal complex and infrequent attack types, as like as R2L and U2R intrusions. The subtle patterns of such sophisticated threats usually cannot be disclosed by standalone models. With a high recall of this hybrid model, it will provide real detection of such kinds of attacks, improving overall security. Moreover, the extraordinary precision of the

proposed model ensures very insufficient false positives, indicating that flagged alerts are actually malicious. This competence is of special worth in IoT environments, as frequent false alarms can saturate security teams and disrupt the normal operation of devices.
The hybrid model also shows adaptability to IoT-specific challenges. IoT networks are categorized by heterogeneous traffic, periodic device behaviors, and resource constraints. The hybrid architecture is robust sufficient to handle these complexities, ensuring reliable detection across diverse IoT ecosystems, including smart homes, healthcare systems, and industrial IoT networks. Its scalability makes it appropriate for protecting interconnected devices in real-world applications.

However, the proposed hybrid CNN-RNN has a limitation that has to be mitigated so that it be able to realistically be implemented in real applications. In form of the computational burden, a major challenge will arise: it has higher computation complexity when compared with standalone models, while the hybrid architecture has especially inherent sequential processing in RNN-LSTM layers. This increases the processing time, which becomes prohibitive for real-time distribution of resource-constrained IoT settings or, in general, at edge or battery-operated nodes.
Another limitation is scalability in large-scale IoT network with real-time requirements. High inference latency and memory consumption may reduce the model's responsiveness for time-critical applications, like healthcare or industrial control systems. Moreover, the model relies on pre-trained features, which may limit its adaptability to novel threats or changing traffic patterns, hence requiring periodic retraining for effectiveness.

In the future, further research should be done by optimizing the model for real-time performance. Techniques such as pruning, quantization, and knowledge distillation can reduce size and computational overhead without loss in accuracy. These optimizations would enable deployment on end devices and improve real-time inference performance. This could be further enhanced with dynamic and adaptive learning mechanisms, such as online learning or reinforcement learning, to create the model adapt to evolving attack patterns and dynamic traffic conditions.

Testing the proposed model for numerous IoT ecosystems, from smart cities to healthcare to industrial IoT, would definitely be of great value in providing much-needed insight into its success in real-world systems. Each domain has its different challenges, like very stringent latency requirements in healthcare, while smart cities have high-volume traffic. Meeting these requirements will increase the utility of the model and its impact as well. Besides, the combination of adversarial training techniques would enhance the model's robustness against sophisticated evasion techniques, such as adversarial examples or traffic mimicry, making it work in highly adversarial settings.
Finally, explain ability and confidence are key to operationalization. Developing interpretable AI techniques for the proposed hybrid model would enable security teams to understand exactly why something was detected, allowing quicker and more informed responses; this would foster trust in the system and drive its incorporation into existing IoT security mechanisms.

C. Comparative Analysis

CNN-RNN hybrid model performance was systematically compared with both machine learning methods and standalone deep learning models on both NSL-KDD, and IoT-23 datasets. This will bring forward the major advantages of this hybrid approach in tackling the new challenges facing IoT malware classification.

a. Advantages of Traditional Methods

Machine learning approaches that have historically been used in malware detection include Decision Trees, SVMs, and Random Forests. These traditional methods function well in less complex categorization tasks. Conversely, they are very reliant on manually crafted feature engineering that requires domain expertise and leads to typically suboptimal performance of complex datasets. In general, classic models cannot generalize well, especially in real-world tasks with constantly changing and progressive malware.

In distinction, the hybrid CNN-RNN model automates the process of feature extraction using CNNs, which can capture the spatial patterns straight from raw networks traffic. For example, CNN layers identify important features like abnormal packet size, protocol usage, and byte distribution without human intervention. Moreover, the combination of LSTM layers within the proposed hybrid model enables it to spot temporal dependencies seamlessly, which is very crucial in the detection of multi-stage or slow-developing attacks like DDoS. This combination of spatial learning and temporal learning gives the anticipated hybrid model a distinct advantage over approaches, permitting it to analyze complex traffic behaviors more effectively and adjust to new threats.

b. Advantages of Standalone Deep Learning Models

Standalone deep learning model includes CNNs and RNN-LSTMs, which have shown great promise in malware detection. CNNs have demonstrated to be very useful in identifying spatial patterns, like packet anomalies and irregular traffic distributions, thus being

very effective in spotting static threats. On the contrary, RNN-LSTMs are good at modeling sequential dependencies, which are vital for understanding time-series data and capturing evolving attack behaviors.

However, all these individual models face their limitations when its usages with the mixed characteristics of IoT traffic, which usually includes static portion and dynamic portion. The proposed hybrid CNN-RNN model overcomes such limitations by exploiting strengths from both architectures. While the CNN part powerfully processes the spatial features, the RNN-LSTM layers analyze temporal dependencies between events, resulting in:

- Superior accuracy for different types of attacks, including multi-stage intrusions and stealthy malware.
- balanced precision rate and recall rate, ensuring less false positives rate and more comprehensive threat detection.

As these two approaches are combined in this model, it shows excellent robustness to heterogeneous IoT-traffic analysis, and also greatly improves the performance in difference to independent models.

c. Key Metrics Across Datasets

This hybrid model also outperforms the traditional methods and standalone deep learning model in all metrics and shows its capability of handling complex IoT traffic. The comparative performances of the methods, standalone deep learning techniques, and hybrid CNN-RNN model on NSL-KDD and IoT-23 datasets are presented in Table 6.

Table 6: hybrid CNN-RNN model on the NSL-KDD and IoT-23 datasets

Model	*Dataset*	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Traditional Methods	NSL-KDD	~85%	~80%	~83%	~81%
Standalone CNN	NSL-KDD	97.8%	95.5%	96.2%	95.8%
Standalone RNN-LSTM	NSL-KDD	96.5%	94.0%	93.8%	93.9%
Hybrid CNN-RNN	NSL-KDD	99.2%	98.5%	98.3%	98.4%
Hybrid CNN-RNN	IoT-23	99.7%	99.2%	99.1%	99.2%

d. Key Observations

These experimental results highlight how the hybrid CNN-RNN model is superior in tackling spatial and temporal complexities in IoT malware detection. Because of the combination of the spatial feature removal capability of CNN with the sequential pattern recognition of RNN-LSTM, improvements are consistent through all metrics and datasets in the anticipated model. It provides the hybrid model with a unique dual capability of not only detecting static anomalies, like abnormal packet sizes, but also evolving attack patterns like multi-stage intrusions or botnet behavior. The superior accomplishment on the NSL-KDD datasets (99.2% accuracy) and IoT-23 dataset (99.7% accuracy) underlines the strong point of this model, which is robust and able to adjust to various environments of traffic and attack types.

Another critical strength of the hybrid architecture is scalability and flexibility. It works well in traditional intrusion classification scenarios and excels in IoT-specific contexts that generally involve unique traffic patterns, low-bandwidth communication, and periodic device behavior. This will guarantee that the proposed model is not limited to a single domain; thus, it is suitable for implementing in varied IoT ecosystems, including smart homes, industrial IoT, and healthcare systems. It also proves that is best model for any challenge on modern IoT malware detection by offering a quite balanced performance between precision, and recall for all classes in general, for complete threat detection while curtailing the rate of false positives of crucial real-world applications. These results are in general proving that this hybrid model is well suited for both present and evolving cyber threats. In the future, real-time application optimization, scalability, and deployment in various IoT environments will be the next steps for solidifying this model as a foundation stone of IoT security.

V. Discussion

Experimental results outline the significant benefits of a hybrid CNN-RNN model in malware classification for IoT environments. This hybrid model will integrate the spatial features removal capability of CNN amid the temporal dependencies modeling provided by RNN-LSTM to offer an in-depth network traffic analysis. This dual focus allows the classification to spot both static anomalies, such as packet sizes that are out of range, and sequential behaviors, such as multi-stage attacks, yielding stronger performance related to machine learning methods and deep learning approaches in isolation. Some of the primary significant strengths of the hybrid model is its high generalization capability. It achieves 99.2% accuracy rate on the NSL KDD datasets and 99.7% on the IoT-23

datasets, hence showing its generalization over various datasets and different types of attacks. The NSL KDD dataset contains traditional intrusion categories like Probe, DoS, and U2R, while the IoT 23 dataset focuses on IoT-specific threats, such as botnet and DDoS attack. The fact that the anticipated hybrid model is able to adapt itself to both traditional intrusion detection and IoT malware detection robustness and its versatility.

Another critical insight is how it can also detect emerging threats. Many IoT-specific attacks, like credential brute-force and DDoS, are staged over longer time frames and thus may remain undetected by any model using purely spatial features. By adding RNN-LSTM for temporal sequence learning to this hybrid model, these subtler, evolving patterns will be learned, thereby increasing the excellence of threat detection. This could be crucial in recognizing furtive, multi-stage attacks that a more straightforward model is unable to recognize.

The high performance of this hybrid model for the IoT-23 dataset pinpoints it for IoT-specific applications. IoT environments are typically characterized by periodic communication patterns in devices due to limited bandwidth and specific scenarios of use. These very characteristics make the traditional systems for detection a bit tricky. This model effectively incorporates spatial and temporal insights into view to allow for efficient discrimination of normal and abnormal traffic. This makes it remarkably suitable for the protection of IoT network in realistic scenarios, such as smart homes, healthcare, and industrial IoT.

Despite the many advantages of the hybrid CNN-RNN model, challenges and limitations also exist. One primary limitation is the computational complexity of the proposed model. Combining the multi-layer convolution operations of CNN with sequential processing in RNN-LSTM increases computational overhead differentiating to standalone models. While this complexity impacts the model's accuracy, it may pose some challenges in real-time deployment on resource-constrained IoT devices such as battery-powered sensors or edge devices with low processing capabilities.

Another challenge is scalability, especially when IoT ecosystems are to scale to millions of interconnected devices. Applications that require large-scale deployment, such as smart cities or industrial IoT networks, have to be responsive in real time and thus require low-latency processing. This may hamper the capability of the intended hybrid model in such scenarios and will require further optimization for practical applicability.

Future Directions

In the forthcoming, efforts should be directed at optimizing the hybrid model for practical applications. One potential line of research is model compression, including pruning and quantization, which reduces both the model size and computation involved. These methods can enable real-world deployment on resource-constrained end devices while maintaining high object detection accuracy.

Another area of improvement is real-time inference. Research on lightweight neural network designs or edge computing frameworks could greatly improve the model's responsiveness, making it appropriate for time-sensitive IoT applications. Accelerating the inference times would ensure the model's effectiveness in dynamic and large-scale environments. Another critical direction is adversarial resilience. IoT malware is continuously evolving, and attackers increasingly use methods designed to evade detection. By integrating adversarial training methods into the design of the hybrid models, it also can be further enhanced to be robust against sophisticated attack strategies, ensuring its reliability in adversarial settings.

Finally, the testing of the hybrid model in different IoT ecosystems is necessary for assessing its flexibility and success in various contexts. IoT systems range from latency-sensitive operations in healthcare devices to high volumes of traffic in smart city infrastructures. The application of the model in varied environments will help the researchers to identify domain-specific challenges and tune the model accordingly, thus enhancing its practical applicability.

VI. Conclusion

This paper proposes a hybrid deep learning approach that fuses CNNs for spatial features removal with LSTMs network for temporal sequence modeling. The proposed architecture is constructed to address unique challenges in IoT malware detection, such as high accuracy for diverse attack types, adaptability to heterogeneous IoT circumstances, and the capability to detect evolving threats. This is reached by a detailed assessment of the anticipated model in being an effective solution for current IoT cybersecurity challenges. The hybrid CNN-RNN model outperforms the traditional methods and standalone CNN or RNN-LSTM architectures. It achieves an accuracy rate of 99.2% on the NSL_KDD dataset and 99.7% on the IoT_23 dataset, thus showing its generalization capability across different types of attack scenarios. Complementary measurement metrics that further demonstrate its success in spotting both unknown and known malware threats. These results reflect the model's capability to provide secure and reliable protection in the ever-increasingly complex IoT ecosystem.

The anticipated hybrid model detects threats comprehensively by leveraging CNNs for spatial features analysis and RNN-LSTMs for temporal dependencies.

This dual focus allows the detection models of a different range of malware behaviors-from instantaneous attacks like network scanning to slow-developing threats such as DDos attacks. Thus, it is capable of handling such scenarios for robust and scalable protection in IoT network.

Moreover, the model has shown pretty strong IoT-specific relevance by showing accomplishment on the IoT_23 datasets. IoT networks normally exhibit a limited bandwidth, periodic nature of device communication, and various forms of traffic flow-all difficult to handle for traditional malware detection models. The hybrid model, in such an environment, turns out to be excellent with precise threat detection and reduced false alarms, which is considered extremely important in real-world IoT applications. Despite its impressive performance, the computational complexity of the hybrid CNN-RNN model points toward a number of future optimizations to ensure its practicality for real-time applications and large-scale deployments.

One of the focus points is real-time applications wherein techniques such as model pruning, quantization, and integration with edge computing reduce the computational burden and make models deployable on resource-constrained IoT devices. Acceleration of inference time will be performed to guarantee the model's feasibility for time-sensitive IoT scenarios.

The other very promising direction is wider application within a diversity of IoT contexts. Expansion to other areas like healthcare and smart cities will demonstrate the adaptability and execution of the model. As an example, it would secure the medical IoT in healthcare; thus, ensuring an integrity of critical patient data and systems. It will help run traffic and energy infrastructure in smart cities while keeping it safe from cyber threats. Another critical area for improvement is adversarial robustness. As cyber threats continue to evolve, it is expected that attackers will adopt techniques that can bypass detection. Research into adversarial training methods will strengthen the hybrid model's resilience, hence its reliability in operational environments. This will address potential vulnerabilities and improve the model's applicability in adversarial settings.

Finally, integrating federated learning approaches can enable decentralized training, where distributed IoT networks collaborate to advance model performance while preserving data privacy. This method will increase the model’s expandability and adaptability, especially in environments with strict privacy regulation. It shows a significant advancement in the field of IoT malware detection by hybrid CNN-RNN, which offers an end-to-end, multi-scale, spatial learning and temporal learning for comprehensive threat analysis. The strong performance in all key metrics, adaptability to diverse IoT environments, and complex and emerging threat detection make it the cornerstone for modern IoT network security. While further optimization is required for real-time applicability and flexibility, the proposed research directions provide evidence of the model's potential for deployment in critical IoT ecosystems. As IoT networks grow larger and more complex, this hybrid CNN-RNN model lays a good foundation for future advancements in cybersecurity.

References

[1.] M. El-Hajj, “IoT Growth and Security Concerns,” *IoT Analytics Journal*, 2017.

[2.] S. Sathyadevan, “AI in IoT Malware Detection,” *AI and Network Security*, 2018.

[3.] W. O'Sullivan, “IoT Security Challenges and Solutions,” *Cybersecurity Review*, 2020.

[4.] T. Nguyen and H. Le, “Deep Learning for IoT Malware Detection,” *Journal of Network Security*, 2022.

[5.] I. Ullah, et al., “IoT Malware Detection using CNN and GRU,” *2021 IEEE International Conference on Computer Science and Network Technology (ICCSNT)*, 2021.

[6.] M. Alazab, et al., “A Survey on Machine Learning for Malware Detection in IoT Networks,” *Future Generation Computer Systems*, 2019.

[7.] I. Haq, et al., “A Multi-Vector Hybrid Deep Learning-based Malware Detection System,” *IEEE Access*, 2021.

[8.] Y. Chen, J. Wang, and L. Li, “A Review of Intrusion Detection in the Internet of Things: Challenges and Solutions,” *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 10312-10328, 2021.

[9.] F. Zhang, et al., “IoT Security: Current Status and Future Directions,” *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2405-2422, 2020.

[10.] D. Zarpelão, et al., “A Survey on Botnet Detection,” *Computer Networks*, vol. 57, no. 2, pp. 1186-1198, 2013.

[11.] H. Alzubaidi, et al., “A Comprehensive Survey on Malware Detection Techniques in IoT Environments,” *IEEE Access*, vol. 8, pp. 186953-186971, 2020.

[12.] N. D. M. Hoang, et al., “A Survey of Machine Learning Techniques for Malware Detection in IoT Systems,” *IEEE Transactions*

on Information Forensics and Security, vol. 16, pp. 2740-2759, 2021.

[13.] J. Li, et al., "Deep Learning for Cyber Security: A Survey," *Journal of Information Security and Applications*, vol. 50, pp. 102-109, 2020.

[14.] S. S. R. Ananthanarayanan and R. B. R. Sinha, "IoT Malware Detection: A Deep Learning Approach," *2020 IEEE International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 1-6, 2020.

[15.] H. M. A. Mohsen, et al., "Detection of IoT Malware Using Deep Learning Techniques: A Comprehensive Review," *IEEE Access*, vol. 9, pp. 102507-102523, 2021.

[16.] Z. Yu, et al., "Anomaly Detection in IoT Networks Using Deep Learning," *Future Generation Computer Systems*, vol. 108, pp. 160-168, 2020.

[17.] K. Shafique, et al., "A Hybrid Approach for Malware Detection in IoT: Combining Machine Learning and Deep Learning," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4919-4927, 2020.

[18.] L. Wang, et al., "IoT Security and Privacy: A Survey on Attack and Defense Mechanisms," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 1-14, 2021.

[19.] J. Zhang, et al., "Malware Detection in IoT Devices Based on Deep Learning Techniques," *IEEE Access*, vol. 9, pp. 106831-106842, 2021.

[20.] A. Alhassan, et al., "Deep Learning for Cybersecurity: A Comprehensive Review," *IEEE Access*, vol. 9, pp. 182926-182948, 2021.

[21.] T. S. O. Akinola, et al., "An Efficient Deep Learning Model for Malware Detection in IoT Networks," *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 234-239, 2021.

[22.] H. H. A. El-Maawali, et al., "Deep Learning-Based Malware Detection in IoT Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 164771-164786, 2021.

[23.] A. Kumar and A. K. Singh, "Recent Advances in IoT Malware Detection: A Comprehensive Survey," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1-36, 2021.

[24.] M. M. Rehman, et al., "Advanced Deep Learning Techniques for IoT Malware Detection: A Review," *Sensors*, vol. 21, no. 4, pp. 1145, 2021.

[25.] C. Liu, et al., "Federated Learning for IoT Security: A Review," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1105-1120, 2021.

[26.] Er, T., et al. (2024). "A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection." IEEE Transactions on Network and Service Management.

[27.] Chen, L., et al. (2023). "Machine Learning-Based IoT Malware Detection Using Statistical Features." Journal of Network and Computer Applications.

[28.] Singh, A., et al. (2023). "Enhancing IoT Security with Ensemble Learning Techniques." Computers & Security.

[29.] Li, J., et al. (2023). "Feature Selection and Machine Learning for IoT Malware Detection." Information Sciences.

[30.] Ahmed, R., et al. (2023). "Anomaly-Based Intrusion Detection in IoT Using k-Nearest Neighbors." IEEE Access.

[31.] Zhang, X., et al. (2023). "IoT Malware Detection Using Decision Tree and Genetic Algorithm." Future Generation Computer Systems.

[32.] Patel, M., et al. (2023). "Support Vector Machine-Based Intrusion Detection for IoT Networks." Computers & Electrical Engineering.

[33.] Wang, F., et al. (2023). "Random Forest-Based Detection of IoT Botnet Attacks." Journal of Cybersecurity and Privacy.

[34.] Kumar, P., et al. (2023). "Naïve Bayes Classifier for IoT Malware Detection." Ad Hoc Networks.

[35.] Rahman, S., et al. (2023). "Intrusion Detection in IoT Networks Using Logistic Regression." Sensors.

[36.] Er, T., et al. (2024). "Deep Learning for IoT Malware Detection: A Survey." IEEE Communications Surveys & Tutorials.

[37.] Chen, H., et al. (2023). "CNN-Based IoT Malware Detection Using Network Traffic Analysis." Neurocomputing.

[38.] Nguyen, T., et al. (2023). "Anomaly Detection in IoT Networks Using LSTM Models." IEEE Internet of Things Journal.

[39.] Haq, Z., et al. (2023). "IoT Malware Classification Using Deep Autoencoders." Journal of Parallel and Distributed Computing.

[40.] Ali, F., et al. (2023). "A Deep CNN Approach for IoT Intrusion Detection." IEEE Transactions on Dependable and Secure Computing.
[41.] Rahman, A., et al. (2023). "IoT Malware Detection with Bidirectional LSTM Networks." Applied Intelligence.
[42.] Chen, Z., et al. (2024). "Hybrid CNN-RNN Architectures for IoT Malware Detection." Journal of Information Security and Applications.
[43.] Smith, R., et al. (2023). "IoT Botnet Detection Using Deep Recurrent Neural Networks." IEEE Transactions on Information Forensics and Security.
[44.] Wang, H., et al. (2023). "GAN-Based IoT Malware Detection Framework." Pattern Recognition Letters.
[45.] Zhao, X., et al. (2023). "IoT Intrusion Detection Using Deep Neural Networks and Attention Mechanisms." Expert Systems with Applications.
[46.] Er, T., et al. (2024). "Hybrid CNN-LSTM Model for IoT Malware Detection." IEEE Transactions on Emerging Topics in Computational Intelligence.
[47.] Chen, Z., et al. (2023). "IoT Malware Detection Using CNN-GRU Hybrid Networks." Computers & Security.
[48.] Nguyen, T., et al. (2023). "A Hybrid Deep Learning Framework for IoT Intrusion Detection Using CNN and Bi-LSTM." IEEE Access.
[49.] Haq, Z., et al. (2023). "Efficient Hybrid IoT Malware Detection Using CNN and Attention-LSTM." Future Internet.
[50.] Smith, R., et al. (2023). "Multi-Layer Hybrid Neural Networks for IoT Botnet Detection." ACM Transactions on Internet Technology.
[51.] Rahman, A., et al. (2023). "IoT Malware Detection with Hybrid Deep Learning and Feature Engineering." Sensors.
[52.] Chen, Z., et al. (2024). "Hybrid Deep Learning Models for IoT Malware Detection in Edge Environments." IEEE Transactions on Cloud Computing.
[53.] Patel, M., et al. (2023). "IoT Malware Detection Using CNN-LSTM with Data Augmentation." Journal of Ambient Intelligence and Humanized Computing.
[54.] Wang, H., et al. (2023). "Hybrid CNN-LSTM Model for IoT Intrusion Detection with Federated Learning." Ad Hoc Networks.
[55.] Zhao, X., et al. (2023). "Attention-Based Hybrid Deep Learning for IoT Malware Detection." Information Sciences.