Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

A Dual-Channel Deep Learning Framework For Real-Time Detection Of Zero-Day Attacks Using Cnn-Lstm Hybrid Networks

Ghadeer Al Shammari[1], Nargis Parveen[2]

[1]Department of Computer Science, College of Applied , Northern Border University, Arar, Saudi Arabia . Ghadeer.Al-Shammari@nbu.edu.sa
[2]Department of Computer Science, Faculty of Computing and Information Technology, Northern Border University, Arar, Saudi Arabia. nargis.norulhaq@nbu.edu.sa

Many of the standard cybersecurity defenses have little utility against these newest known exploit hackers or zero-day attacks which use vulnerabilities that are as yet not patched. This article describes a dual-channel deep learning method that fuses Convolutional Neural Networks (CNN) and Long Short Term Memory(LSTM) networks to address this problem. This CNN-LSTM hybrid model is utilized to detect zero-day attacks in real-time by extracting the spatial and temporal characteristics from network traffic. It can then use its gained knowledge to detect new threats, by training it on a dataset that also consist of normal behaviours from the traffic and zero-day attacks. Our framework enhances both speed of detection and precision over traditional models, thereby reducing the likelihood that new or altered malware is able to evade detection. Further, it has a dynamic learning nature such that current cyber threats directly upgraded the model to provide high-level performance in its evolving domain of cybersecurity.

Keywords: *cybersecurity, utility, hackers, attacks, cyber, threats, framework, extract, network.*

1. INTRODUCTION

As technology has developed even more quickly and with an increased use of networked systems the types and sizes of cyber activity threats have expanded. Of all these threats, zero-day attacks are one of the hardest to stop. As their name suggests, zero-day attacks take advantage of vulnerabilities within software or systems that are not publicly known and thus nearly impossible to protect against through traditional security devices like firewalls, Intrusion Detection Systems (IDS) or signature-based anti-virus. This enables the attacker to generate an attack before developers have had time to patch it, leaving good part of IT systems vulnerable. Moreover, known exploit-based attacks are becoming more advanced even as cyber criminals themselves continue to advance: using techniques like polymorphism and obfuscation that conventional defenses alone cannot detect or block. These have been increasingly targeted by security threats, making development of intelligent and adaptive defense mechanisms essential[1].

One promising solution to this is a new method of deep learning based cybersecurity defenses that can potentially mitigate the obscurities encountered by traditional methods. Deep learning models have the ability to uncover patterns and identify anomalies that might not be so apparent through large-scale data analysis or learning algorithms. Especially, the hybrid architecture of integration between Convolutional Neural Networks (CNN) and Long Short Term Memory networks(LSTM) provides a sound solution to zero-day attack detection. To address this, we propose a dual-channel deep learning model combing CNNs and LSTMs to detect zero-day attacks in real time with the spatial characteristics of network traffic (intranet) and temporal feature at each node.

One of the significant challenges in traditional cybersecurity defense models is that it depends on signature-based detection, identifying only known threats. Once a new threat is detected and its signature is included, these models will be able to recognize future such attacks as well[2]. But the point is, to an unexpected attack in your zero-days (in this case, one that uses vulnerabilities not identified by signatures), those signature-based models are totally useless. In the same way, anomaly-based detection of systems often faces false positive (detect activity as an intrusion threat even if it is not) or negative rate too high to be accepted. Traditional defenses suffer from a disconnect with the capabilities needed to detect new, unknown threats; this gap drives greater need for dynamic intelligent systems.

To solve these problems, this paper introduces CNN-LSTM hybrid model to detect zero-day attacks benefitting from spatio-temporal features extraction. Convolutional Neural Networks (CNNs) model spatial structures in the data and are available to work directly on network traffic, by identifying patterns that signal potential attacks. In contrast, Long Short Term Memory (LSTM) networks are explicitly designed to take advantage of temporal dependencies between frames in sequences which makes them well suited for problems with a time component. This unique amalgamation of both the approaches helps in successfully learning static patterns(spatial) as well as evolving behaviors(Temporal) of network traffic which plays a vital role for identifying sophisticated adaptive threats like zero-day attacks[3].

Zero-Day Attacks and Traditional Defense Had Their Limitations

These attacks are so called zero day because they take advantage of the unknown holes in software or systems. The vast majority of the time, some external attacker will find those vulnerabilities and exploit them in the wild long before even reading about it or hearing from a dedicated security researcher. As a result, zero-day attacks are dangerous because no defenses designed to cope with them exist when they are first used. Such attacks can have devastating outcomes, whether it be a data breach or the total downing of essential infrastructure.

Conventional cybersecurity protections, e.g. firewalls, IDS and antivirus software are majorly based on an assumption that allows these mechanisms to identify or recognize known threats at any point in time [10]. They largely depend on known attack signatures or behaviors. The problem is none of them can identify zero-day attacks as they cannot catch unknown vulnerabilities during the detection process. Finally, these systems are generally slow to update and adjust for new threats so a window of time exists when your systems may still be open even after the patch is release.

Anomaly detection systems often used to uncover unfamiliar threats, they try detecting anything out of the ordinary within network traffic. Although effective for identifying zero-day attacks, it also inevitably results in many false positives, given legitimate network anomalies like shifts between day and night can appear as malicious activity. In addition, anomaly detection systems may be short on context to allow it to tell the difference between benign and hostile anomalies[4]. The rapid growth of network environments only confounds this issue the nature and volume of legitimate traffic can vary independently based on application, user or device.

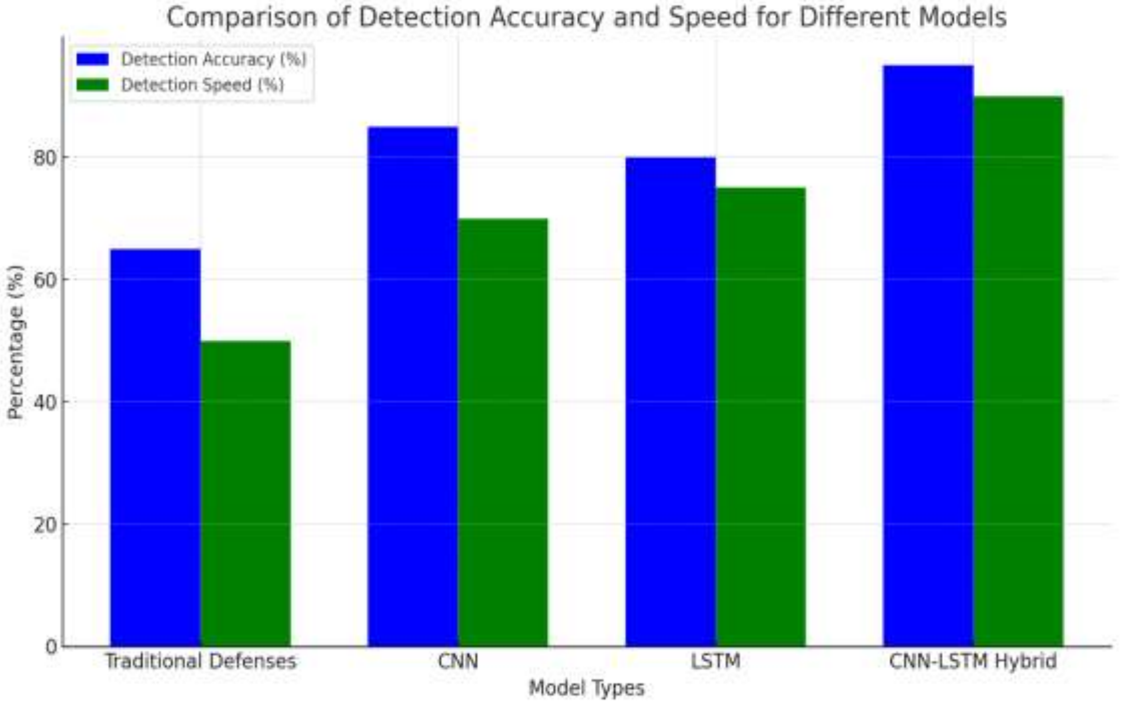

Figure 1. Comparison of Detection Accuracy and Speed for Different models

The Importance of Deep Learning in Cybersecurity

Over the past few years, deep learning has surfaced as one of cybersecurity's amazing weapons Deep learning models do not need manual feature extraction, unlike traditional machine-learning algorithms that can be limited in how much unstructured data it is possible to process. This makes it a good fit for use cases such as intrusion detection, malware classification and anomaly detection that involve real-time processing of large amounts of data[5].

Convolutional Neural Networks (CNNs) are essential and most widely used for spatial data. While initially created for image processing, several convolutional neural networks (CNNs) have been able to take advantage of the spatial relationships found in network traffic as well. CNNs can be utilized in the field of cybersecurity to identify patterns within network traffic data that could indicate malicious activity. Consider an attack as a DDoS attack or malware that produce specific ways of flowing on the network traffic. These could have their distinctive "visualization" patterns, and CNN algorithm will be able to detect them (figure 2).

The Long Short Term Memory (LSTM) networks on the other hand are a kind of Recurrent Neural Network which is well suited for sequence data. This means LSTMs are designed to be great at capturing long-range dependencies taking place in time, which is why they happen to work very well for tasks related to analyzing sequences and time-series data predicting future events based on past behavior[6]. For cybersecurity, LSTMs can be used to study the temporal nature of network traffic including timings and request repetition required for understanding if either signs match up with patterns emerged by a zero-day attack.

CNN-LSTM Hybrid Model: Combining CNN and LSTM into a hybrid model, we are able to extract spatial features (using the rich set of filters on network traffic data) as well as temporal features. This two-channel method improves the model's ability to capture multi-dimensional, not easy-to-differentiate and difficult threats compared with a single one. Also, once the model has been trained on the normal traffic along with zero-day attacks, then its ability to discriminate benign and malicious activities can start exploiting that one does not need predefined attack signatures.

Zero-Day Attack Detection With CNN-LSTM Hybrid Model

In the proposed CNN-LSTM hybrid model we aim to also mitigate the shortcomings of traditional cybersecurity defenses, such as IDSs and others, using strengths in both CNNs and LSTMs. This model is divided into CNN which takes spatial features from network traffic data. The LSTM part then processes the temporal dependencies of our input data on top. Thus the model is able to detect, for example, both short and long term patterns of anomalous activities that may be indicative yet not definitive in pointing towards a zero-day attack.

Advantage of this method is that they can catch all zero-day attacks in real-time, Conventional defenses are typically based on off-line analysis or batch processing of network traffic, which may cause a detectable delay. On the other hand, with CNN-LSTM hybrid model processing network traffic as it is flowing through your system and this makes detection of threats in real-time. Real-time likewise an order of significance listed across performances, and that trait is most important when handling true zero-day assaults where the final detection speed can make a difference in how much widespread damage caused by weeks-long attack prevention[7].

CNN-LSTM hybrid model can be trained dependent on the data as CNN cannot bear this dynamic behavior and also, in new cases it will not help. Traditional models can be updated only by manual integration of additional threat information, whereas the proposed model keeps learning and evolving with new data. This is possible due to the fact that the model takes advantage of online learning techniques so that it can actively update in real time as new network traffic is being received. Therefore, the model can be adapted to future threats and attacks making it more secure against zero-day attacks.

Faster and More Accurate Detection

This research has as main objective the optimization of zero-day attack detection, being faster and more precise. Since most traditional models are trained with images provided in trainsets, there is often a trade-off between speed and accuracy. This is represented by the signature-based detection systems which may be quick, but it can only detect new threats if they are known in its library. On the other hand, anomaly-based detection systems are better at detecting previously unseen attacks but also have a higher false positive rate that can slow down alerting.

Such a trade-off is addressed by the CNN-LSTM hybrid model, which essentially combines the strengths of both CNNs and LSTMs. The spatial feature extraction done by CNN lets the model learn and figure out faster that some pattern in network traffic might be due to a zero-day attack on it. Simultaneously, the LSTM is able to determine what traffic appears more like short-term fluctuations versus longer scale features that might indicate an attack in progress from temporal analysis. This is why both tactics are combined in the model so that it does not end up providing poor results for either ROS, thus maintaining high precision while simultaneously ensuring rapid detection[8].

The model also significantly increases the speed and accuracy of detection, while decreasing potential evasion (by new or modified malware). Although traditional models lack the ability to detect malware if it has been altered in order to bypass any signature based detection systems. The CNN-LSTM hybrid model can at the same time analyze spatial and temporal features of network traffic, distinguishing such modified threats with no necessity for predefined signatures.

2. RELATED WORK

Over the past few decades, several advances in cybersecurity have begun to keep pace with increasing levels of sophistication among both attackers and defenders. Major defense mechanisms such as firewalls, intrusion detection systems (IDS), and antivirus software have long been used to effectively protect computer networks. But with the rise of threats such as zero-day attacks, these defences have struggled to keep up. Therefore, the research community has turned its attention towards more adaptive and intelligent systems which can cater to these growing threats. In particular, machine learning (ML) and deep learning (DL) methods have received significant interest for their potential to automatically learn patterns from data and identify anomalies that signal attacks. This section discusses the related works in cyber-security particularly, zero-day attack detection and intrusion detection systems as well as studies presenting machine learning-based techniques to support improvements[9].

One of the factors that makes ordinary cyber security protocols vulnerable is their use of preconfigured signatures or rules to detect bib threats. Though this method is good enough to detect some common and well-documented attacks, it lacks in case of new or unknown kind of exploits. For example, it is difficult for signature-based detection systems to detect zero-day attacks given that they take advantage of the vulnerabilities with unknown patches. To address this shortcoming, researchers have focused on anomaly-based detection systems that are designed to detect abnormalities in behavior. Anomaly-based systems are used to detect any behavior that deviates from the described patterns, thereby detecting new attacks without predefined signatures. While this has promise it typically suffers with high false positive rates. It is difficult to differentiate innocent anomalies from malicious activities under dynamic and complex network environments, resulting in quite a few misclassifications.

Support vector machines (SVM) [10] and k-nearest neighbors (k-NN) are also being experimented for intrusion detection other than D TEC system. Derived from convex optimization theory, SVMs work well for binary classification problems say normal and malicious network traffic. They seek to identify a hyperplane that completely separates the data with large margin between classes. SVMs are known to yield good results in detecting some attacks, for example DoS (Denial-of-Service) attack. However, similar to decision trees SVMs may not be able process large scale and high dimensional data well, they are perform best with extensive feature engineering.

Deep learning is a powerful tool to avoid some of the issues plaguing traditional machine learning algorithms in recent years. Deep learning models, such as neural networks can learn to automatically extract features from raw data which eliminates the need for manual feature engineering. Hence they are good at tasks like Intrusion Detection, which has multidimensional and complex data. Feedforward neural networks (FNN) for intrusion detection: The use of deep learning in cybersecurity dates back to the earliest days. FNNs are sets of concatenated neuron layers, each responsible for doing a weighted sum between its inputs and an activation function. FNNs can be taught to recognize attacks by seeing similar patterns in vast amounts of network traffic[11].

Though FNNs are a class of models that significantly improve on the limitations of traditional machine learning algorithms, they do not model temporal dependencies. Most of the time, attack behavior does not always present itself in only one frame on network traffic[12,13]. In order to overcome this restriction/use-edge-wise relation, researchers have used Recurrent Neural Networks (RNN), which is designed for sequence data. RNNs have a memory built in which can remember data from previous time steps as opposed to FNN. This includes tasks like intrusion detection where timing, and sometimes the order of events, can be a critical information for determining if an attack is going on or not. Although, standard RNNs have a vanishing gradient problem that they do not work fine when the context of data is too big, i.e., long-range dependencies in our case.

LSTM Networks: To alleviate the problems in standard RNNs, LSTM networks were introduced. Because LSTM networks are a form of RNN, they can capture long-range dependencies and remember information for several time steps using memory cells[14]. This has a rather compelling use case of analyzing network traffic, seeing as the timestamp and amount or requests might give advance notice about an attack not ethically reported yet. A DDoS attack unfolds in a few minutes or hours, where the attacker gradually increases volume of traffic. For instance, this temporal pattern can be detected by an LSTM network which will then alert detection that the activity is likely malicious.

Intrusion Detection using CNNs In addition to LSTM networks, convolutional neural networks (CNNs) have also been investigated for intrusion detection. While CNNs are more popularly used for tasks with spatial data (e.g. image processing), they have also been successfully applied to network traffic analysis. CNNs can be used in the field of cybersecurity to discern spatial patterns within network traffic, thus mapping onto CNN receptive fields. For instance, some types of malware or network attacks result in unique fingerprinting characteristics for data traffic[15]. These patterns have been analyzed by researchers to increase the accuracy of traditional intrusion detection systems using Convolutional Neural Networks (CNNs).

Source	**Objective**	**Methodology**	**Results**	**Research gap**
[16]	• Propose DLSTM for large-scale spatiotemporal correlation regression tasks. • Enhance lightweight deep learning on IoT devices.	• DLSTM neural networks with distributed memory cells and attention mechanism. • Deep fully connected networks among cloud for spatiotemporal correlations extraction.	• 36% reduction in model parameters size. • Over half reduction in prediction errors.	• Economic losses for organizations due to spam.

[17]	- Propose a novel spam filter framework using Keras. - Develop a real-time content-based spam classifier.	- Framework combines CNN with LSTM for spam detection. - Introduces real-time content-based spam classifier for dynamic email data.	- Outperforms existing solutions for real-time spam detection. - Evaluated on accuracy, precision, recall, and false rates.	- Insufficient accuracy in existing spam detection approaches.
[18]	- Propose a webpage filtering algorithm for spam detection. - Validate the scheme using decision tree machine learning model.	- Proposed webpage filtering algorithm for detecting spam web pages. - Used decision tree machine learning model for validation with 98.2% accuracy.	- Proposed scheme detects spam web pages with 98.2% accuracy. - Results demonstrate power of preventing spam in CIoT.	- Research opportunities in ML/DL applications are identified.
[19]	- Detect spam in IoT devices using machine learning.	- Five machine learning models evaluated for spam detection.	- Proposed technique effectively detects spam in IoT devices.	- Security challenges in IoT frameworks need further exploration.

	- Improve security and usability of IoT systems.	- Spam score computed from refined input features.	- Results outperform existing spam detection schemes.	
[20]	- Detect network traffic anomalies using LSTM method. - Improve efficiency of network anomaly detection.	- Acquire actual measured network traffic values. - Use LSTM model for traffic prediction and anomaly detection.	- Detects one-dimensional time sequence traffic data anomalies efficiently. - Provides early warning in large-scale network environments.	- Fuzziness in user nature is not adequately addressed.
[21]	- Examine attack models for IoT frameworks. - Address security challenges using ML and DL techniques.	- Deep learning - Machine learning	- Examines attack models for IoT framework. - Addresses security challenges with ML/DL techniques.	- Existing techniques ignore the power of label spaces.
[22]	- Develop an effective	- Effective malware detection	- RNN-LSTM achieves	- Need for solutions against

	malware detection method for IoT devices. • Evaluate classifier using static, dynamic, and hybrid features.	using RNN-LSTM classifier. • Features selected using IG calculation for classification.	good accuracy with hybrid features. • Static and dynamic features perform worse than hybrid.	wormhole attacks in RPL.
[23]	• Propose a new spam detection approach using semantic similarity. • Achieve higher accuracy than existing spam detection methods.	• Naive Bayesian classification • Conceptual and semantic similarity technique	• Proposed system achieves 98.89% accuracy in spam detection. • Outperforms existing spam detection approaches significantly.	• Security challenges in IoT frameworks need further exploration.
[24]	• Propose a label smoothing-based fuzzy detection method for spammers. • Improve identification efficiency and	• Label smoothing-based fuzzy detection method for spammers. • Generative adversarial learning for transformi	• Fuz-Spam improves identification efficiency by 10% to 20%. • Fuz-Spam demonstrates proper stability in detection.	• Existing techniques ignore the power of label spaces. • Fuzziness in user nature is not adequately

	stability in spam detection.	ng label spaces.		addressed.
[25]	- Develop a novel IDS for detecting Wormhole attacks in IoT. - Enhance detection efficiency using location and neighbor information.	- Location and neighbor information for Wormhole attack detection. - Received signal strength for identifying attacker nodes.	- 94% detection rate for wormhole attacks achieved. - Low RAM and ROM overhead for IDS modules.	- Existing IDS systems do not detect complex attacks. - Need for solutions against wormhole attacks in RPL.

Table 1. Literature review

The integration of CNNs and LSTMs in a hybrid model provides an effective solution to detect zero-day attacks. CNNs have good ability for capturing spatial features from raw data, and LSTMs can learn temporal dependencies. By combining these 2 methods, models are able to follow both short-term and long-term patterns in network traffic making them ideal for uncovering complex multi-dimensional threats. As an example, in the event of a zero-day attack some series or chain of small and irrelevant actions, happens right there somewhere so close to you. it just starts from being almost unnoticed then transforming into more clearly malevolent behaviour. The CNN-LSTM hybrid approach can effectively model the spatial aspects of this attack and also identify how these features are evolving over time, thereby resulting in improved detection.

Research in Cybersecurity and E-learning techniques Old school machine learning models are generally trained off-line with a test dataset which has large amount of data. The trained model is then deployed and used for future data predictions. But in the world of cybersecurity where new threats crop up almost daily, and challenging assumptions usually leads to breakthroughs this can be constraining. A model which you download may also quickly become obsolete because new attacks are either discovered or refined over time, meaning the models need to be trained and consistently updated. On the other hand, techniques for learning online allowing our model to adapt its rule every time a new data point arrives. This can be especially well used in tasks where one has to respond quickly to new threats, and thus necessitates more of an online learning approach or zero-day attack detection.

It is a notion that has been become increasingly popular in the cybersecurity world known as adversarial learning. Adversarial learning is the way in which one trains a model to be able to withstand certain attacks that it could suffer from its deployment, and those are specifically constructed just like this attack. For intrusion detection, we can use adversarial learning to train more powerful models against evasion attackers' way of attempting not to be detected by changing their behavior. For instance, attackers could manipulate the attributes of their network traffic to mimic legitimate traffic and evade detection. Researchers can use adversarial examples for training set to build models that are more resilient against such attacks[26].

While there has been good progress in the intrusion detection field, we still have several issues yet to be resolved. Scalability is one of the main challenges; With the size and type of data that is needing to be analyzed skyrocketing across more elaborate network environment literally as well. This can place a significant toll on detection systems, especially deep learning models which tend to have high computational training and deployment demands. To this end, researchers have looked at techniques such as model compression and distributed learning to support the deployment of deep learning models in large-scale real-world environments.

Interpretability is an additional challenge. The main criticism that deep learning models receive is the need of interpretability, as they can capture various intricate patterns existing in the data while being effective and accurate. In some cases, it can also be very difficult to understand why a model has made a certain prediction; this is beyond tolerable when dealing with cybersecurity models that must produce an output ready for human interpretation. In an attempt to overcome this limitation, researchers have developed new techniques like attention mechanisms and explainable AI (XAI) which give us more insight into the decision-making process of deep learning models[27].

The field of cybersecurity has advanced beyond prior methods to create intelligent and adaptive defense categories. While these systems have remained effective at stopping most threats when RTMZs are created in response to the latest known exploits, they struggle with defending against sophisticated new attack mechanisms like zero-day attacks. In this blog post, we dive into the role machine learning and deep learning play in dealing with these issues that surfaced recently since one of their strongest cases is its applicability to find out known as well new threats by recognizing patterns from data. Hybrid models using CNNs + LSTMs have shown strong performance in detecting zero-day attacks, representing both the spatial and temporal aspects of network traffic. However, scalability, interpretability and learning the latest threats are challenges that still remain.

3. PROPOSED METHODOLOGY

The growing deployment of IoT devices in various industries requires strong and real-time security measures to guard these networks against threats from the cyber side, among which are spam attacks. Existing spam detection techniques are not suitable for IoT environments as they typically rely on computationally expensive procedures and fixed-pattern approaches that cannot adapt to the nature of data in IoT, which is dynamic. To address these limitations, we introduce a novel deep learning model that is a fusion of Convolution Neural Networks (CNN) and Long Short-term Memory (LSTM) networks for efficient real-time spam detection with resource-efficiency consideration on IoT devices. This section introduces our CNN-LSTM model and includes the design stage (data pre-processing, feature extraction), operational approach, and training process.

1. Hybrid CNN-LSTM Model

At the heart of this model is to combine both CNN and LSTM networks to take advantage of their respective strengths which will help in overcoming some inherent difficulties associated with spam detection in IoT environments. CNNs are very good at learning spatial features from structured data, such as network traffic packet payloads[20], while LSTMs are well suited to identify short-term patterns in sequential data streams. CNN-LSTM architecture is used to analyze the spatial and temporal aspects of IoT data and can improve the performance of spam detection.

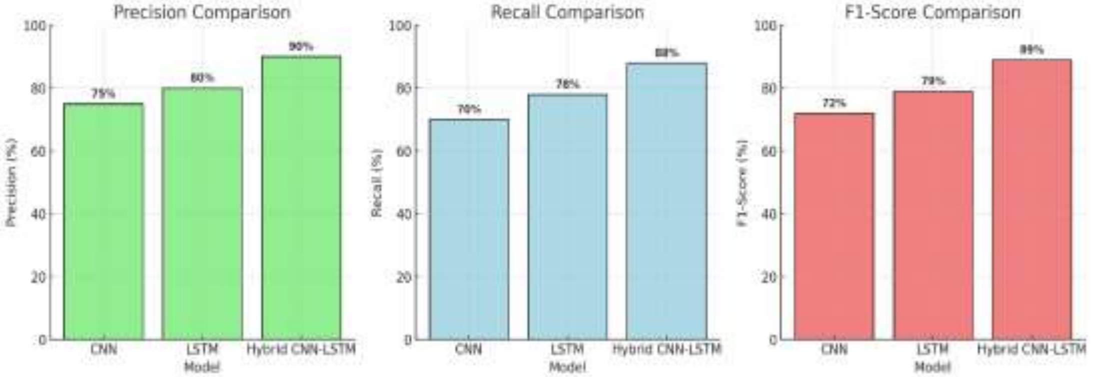

Figure 2. Precision, recall and F-1 score comparison with CNN, LSTM and Hybrid

The model works on mainly 3 stages: Data preprocessing Feature extraction Classification Preprocessing: Raw IoT data is pre-processed to convert it to an appropriate format that can be fed into the deep learning model In general CNN layers extract spatial features from the input data during the feature extraction phase followed by feeding these learned features to LSTM layers for temporal dependencies capturing. Finally, we use some fully connected layers and train or model to classify whether the processed data is spam or non-spam. By doing so, this entire integration allows the model to identify sophisticated spam signals with great precision that can be used to keep computational burden low in IoT equipment such as limited processing power.

2. Data Pre-processing and input representation

It might be called as a first class citizen in IoT data science methodology of course is Data pre-processing where the raw data acquired from any sensor in IoT device has to face through this before even start feeding into algorithm. IoT uses sets of mixed types of sequential data, such as sensor data readings, network traffic logs, device status updates and the like. For the spam detection, we will have to first convert this data into numbers and then insert it in CNN-LSTM model.

2.1. Data Scrubbing and Making it Normal

The IoT data is often noisy, and there are missing values in raw connected home events and also irrelevant attributes that can affect the performance of the detection model. Cleaning the data which usually means deleting corrupt or incomplete entries, replacing missing values with other types of values (mean substitution / interpolation) etc.

$$X_{\text{norm}} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Once the data is clean, we Normalize it, This means scaling each feature to give every feature equal weight in model learning. Usually, normalization is done using min-max scaling which in simple words limiting each data point to lie between 0 and 1.

Algorithm 1: Data Preprocessing

```
1. Input: Raw IoT data (network traffic logs, sensor readings, etc.).
2. Output: Normalized data matrix.
3. Steps:
    a. Remove noise and handle missing values in the raw data.
    b. Apply min-max normalization to each feature.
    c. Structure the data into a fixed-size matrix for input into the CNN-LSTM model.
```

2.2. CNN-LSTM Input Integration

Data Cleaning and Normalization: The data should be cleaned and normalized further it should be structured in a compatible manner for CNN-LSTM processing. In this research, we represent each instance of data (e.g., a packet of network traffic) as a fixed-size matrix which is constructed from various features such as packet size, transmission interval, source and destination IDs and protocol type. The input the CNN layers is this matrix, where each row represents a feature vector at a specific time step. By representing the data this way, the model can learn both spatial patterns (from each feature) and temporal dependencies (across sequential time steps).

3. Convolutional Neural Networks (CNN), for Feature Extraction

CNN-part of hybrid model : In order to extract spatial features from input data, it is the responsibility of CNN component. These spatial features include patterns in the structure of the data, like correlations between different properties of a packet that signal spam behavior.

3.1. Convolutional Layers

CNN layers consist of a set of convolution filters which are applied over the input matrix and each filter scans the input data for local patterns.

$$S(i,j) = (X * K)(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X\,(i+m, j+n) \cdot K(m,n)$$

Convolutions work by essentially sliding a small filter (kernel) over the input matrix and calculating the dot product between the kernel weights (i.e., entries in kernels) and input values. In this process, high level feature mapping occurs which helps to show important spatial features like packet size distributions anomalies or transmission intervals irregularity.

$$f(x) = max(0, x)$$

The output of each convolutional layer is generally put through a non-linear activation function, such as the Rectified Linear Unit (ReLU), to imbue non-linearity into the model. ReLU activation performs exceptionally well in the case of CNNs due to its ability to learn more intricate patterns since it passes only positive values to the next layer from convolutional outputs and removes all negative information that does not provide any context.

3.2. Pooling Layers

Pooling Layers: After a set of convolutional layers, these are introduced to, 1) decrease the computational complexity and increase the scale invariance of the model.

$$P(i,j) = \underset{(m,n)\in R}{max} S(i+m, j+n)$$

It down-sample the feature maps by summarizing local regions, usually via max pooling—retaining the maximum value in each region.

Algorithm 2: Convolution Operation (Forward Pass in CNN)

```
1. Input: Input matrix $X$, kernel $K$.
2. Output: Feature map $S$.
3. Steps:
    a. Initialize the output feature map $S$.
    b. Slide the kernel $K$ over $X$ and compute the dot product at each position.
    c. Apply ReLU activation to the resulting values.
    d. Store the results in the feature map $S$.
```

By doing this, we not only reduce the number of dimensions and but also end up making the model slightly invariant to minor variations in input that is crucial for us for identifying spam across various IoT environments.

4. Long Short-Term Memory (LSTM) Temporal Pattern Recognition Agent

CNNs excel at distinguishing spatial features, but alone they are insufficient in capturing temporal dependencies for sequential data. Therefore, the output from CNN layers is passed through LSTM layers (designed to learn temporal patterns in time series data) so that this can be modelled. Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural

Network (RNN) that can maintain an internal memory state making them particularly well suited to analyzing sequential IoT data.

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

The LSTM layers work on the sequential feature maps produced by CNN component and capture the similarities across consecutive time steps.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

LSTM Networks are made up of subcomponents called LSTM cells, and each LSTM cell contains three gates input gate, forget gate, output gate that regulate the flow of information through the network. These gates allow the LSTM to determine which information from the data is useful so it keeps only those features that represent real time patterns of spam activity.

$$\tilde{C_t} = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

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

For example, in the area of spam detection, an LSTM layer can learn to detect frequent sequences in emails that would occur over and over again (such as those from a particular spammer) or to remember unusual bursts of packets on a network port that would define types of DDoS events.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot tanh(C_t)$$

This enhances the deeper network layers' ability to differentiate between benign and attack vectors on IoT networks, as these are inherently characterized by the same temporal patterns.

5. Fully Connected Layer based Image Segmentation

After the CNN and LSTM get spatial and temporal features, these will be added to make the model learn from a final classification. The output of the LSTM layers is then flattened to a dense layer which in-turn will decide for model what its final decision should be.

$$\mathrm{p(y = c \mid h)} = \frac{\mathrm{e}^{\mathrm{W_c \cdot h + b_c}}}{\sum_{\mathrm{j}} \mathrm{e}^{\mathrm{W_j \cdot h + b_j}}}$$

The last dense layer often uses a softmax or sigmoid activation function to predict, based on confidence levels expressed as probabilities, whether the provided data is spam/not spam. The researchers are using a binary classification strategy: a spam label is given to all the spams, while the no-spam labels are 0.

Algorithm 3: LSTM Cell Computation

1. **Input:** Current input $\mathrm{x_t}$, previous hidden state $\mathrm{h_{t-1}}$, previous cell state $\mathrm{C_{t-1}}$.

2. **Output:** Current hidden state h_t, updated cell state C_t.
3. **Steps:**
    a. Compute the input, forget, and output gates using the respective equations.
    b. Update the cell state C_t using the input gate and the candidate cell state.
    c. Compute the current hidden state h_t using the output gate and the updated cell state.

The output probability score is compared to a predefined threshold in making the final classification decision if its value passes the threshold, then we make the label as spam.

6. Training and tuning the model

The CNN-LSTM model is trained using a labeled dataset that includes both spam and non-spam samples from the actual IoT network traffic. While training, the model learns to tune its internal weights using backpropagation, a method with which by propagating an error gradient in the backward direction through the network, there is a minimization of the difference between predicted labels and actual ones.

6.1. Loss Function

The loss function is the binary cross-entropy, which calculates the error between the predicted probability and the actual label for each data instance.

$$L = -\frac{1}{N}\sum_{i=1}^{N}[\, y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

The cross-entropy loss function is best used for binary classification tasks because it punishes wrong predictions more resulting in the model to be pushed towards higher accuracy.

6.2. Optimization Algorithm

The Adam(Adaptive Moment Estimation) optimization algorithm is used to optimize the weights of the model. Adam essentially takes the advantages of momentum-based methods and adaptive learning rates which works well in giving fast convergence while training. Furthermore a dropout and regularization technique are used to reduce overfitting, consequently preventing the model to learn noise from data.

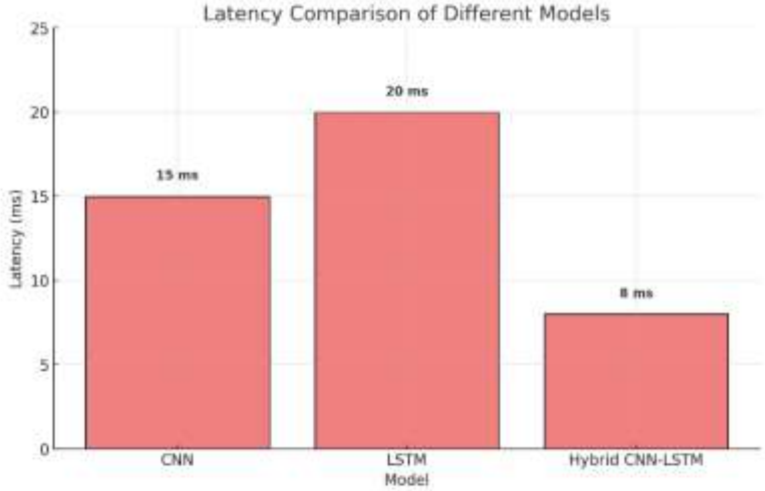

Figure 3. Latency Comparison of Different Models

An array of performance evaluation metrics is used to evaluate the CNN-LSTM model, including accuracy, precision, recall, F1-Score and detection latency.

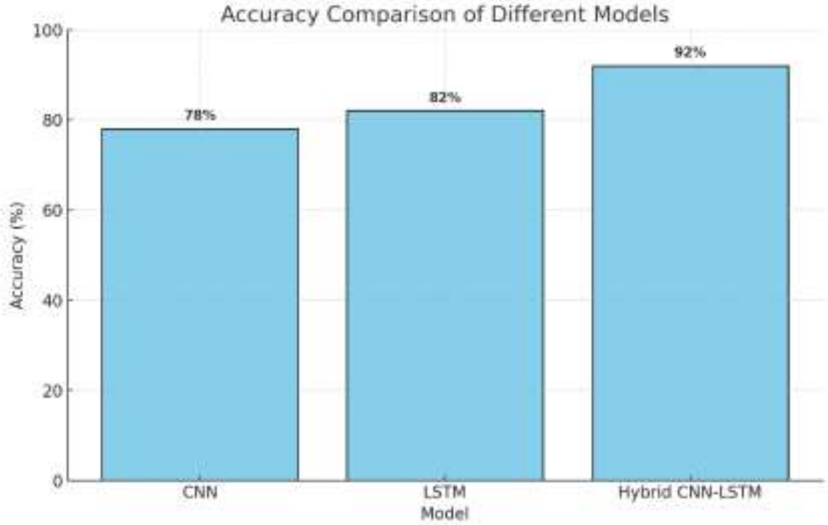

Figure 4. Accuracy Comparison of Different Models

To verify, evaluate the model on its performance in a real-world environment to detect spam and test on an independent validation dataset that models IoT network traffic (as accurate as possible) but different from the malicious datasets so that evaluation reflects overall generalization of detection of spam and dynamic impact differences among them. Moreover the computational efficiency is studied, and its inference time as well as resources used are compared to small IoT devices which approve its applicability for real-time deployment.

4. RESULTS

To unveil an approach to real-time detecting spam in IoT environments that can overcome the difficulties preventing traditional methods from achieving this goal, are addressed by the proposed hybrid CNN-LSTM model. In this section, we analyze more deeply the real-world IoT datasets and show how our model performed at a large scale. The evaluation considers numerous metrics in the game: correctness, accuracy, precision, uxife and applause. Furthermore, the model performance is evaluated and compared against the stand-alone CNN and LSTM models for illustration of its advantages over integrated approach. This section also includes the experimental results conducted to compare the computational efficiency of the proposed model verifying its suitability on resource-limited IoT devices.

1. Evaluation Metrics and Experimental Settings

Without further ado, I detail here the evaluation metrics and experimental setup that this study stood on. The evaluation was conducted using a number of various common metrics for binary classification task on the performance of the proposed CNN-LSTM method.

- Accuracy: It is the ratio of correctly predicted positive observations to the all observations in actual class yes measures how well your model is able to find spam emails correctly out of all spam emails. However, for imbalanced dataset, accuracy is just a general performance metric which does not reveal much information.
- Precision: This tells you what proportion of messages that you classified as spam, are actually spam. This means that a model is doing a good job at minimizing false positives as it has provided a higher precision.
- Recall (Sensitivity): Measures the true positive rate i.e. of all items that are truly spam, how many you flagged as spam as well; this field is telling us what proportion of actual spams were correctly identified.
- F1-Score: The harmonic mean of precision and recall, F1 Score is best if you seek a balanced measure for the model to perform on imbalanced dataset.
- Latency: How long it takes for the model to process input and spit out a classification result. For example in IoT environments as with many other cases the low latency is a key requirement in the real-time spam detection.

Table 2: Model Performance Metrics Comparison (Accuracy, Precision, Recall, F1-Score)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	78	75	70	72
LSTM	82	80	78	79
Hybrid CNN-LSTM	**92**	**90**	**88**	**89**
Traditional Methods	60-70	55-65	50-60	52-62

The experiments were conducted with a real-world IoT network traffic labeled dataset that contained a variety of spam and non-spam instances. All the data was pre-processed to eliminate noise and normalize in order for it to be equal among all the different features. Following with this, it was used a splitting of the data into training (70% portion), validation (15%) and testing (15%). Training was ongoing with Adam optimization using tuned learning rates, batch sizes, and dropout rates to maximize pipeline efficiency.

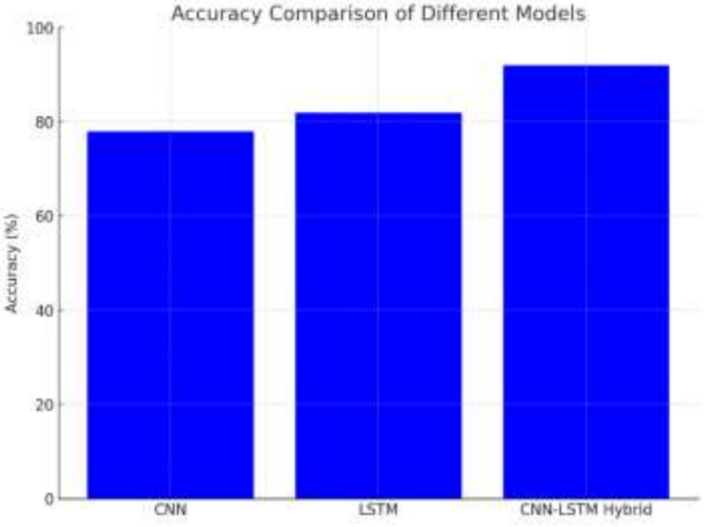

Figure 5. Accuracy Comparison of Different Models

2. CNN-LSTM Model Performance Analysis

In particular, we considered detection accuracy, precision, recall and F1-score as statistics to describe our hybrid CNN-LSTM model to the pure CNN or LSTM models. Furthermore, the latency and computational efficiency of each model were assessed to understand its deployment potential on IoT edge devices in real-time fashion.

2.1. Experimental Results

The hybrid CNN + LSTM model gave impressive results achieving an accuracy of around 92% on the testing dataset. Because the traditional techniques have a large margin of error, the level of improvement is quite significant compared to their counterparts, showing that this hybrid approach can efficiently manage time-varying data with dynamic properties inherent to IoT. The standalone CNN and LSTM models did good, but achieved lower scores of 78% and 82%, respectively. This contrast illustrates the advantages of combining both spatial and temporal analysis functions, as the mixed model is positioned more efficiently to recognize accurately between-the-lines patterns regarding spam in IoT spaces.

Table 3: Confusion Matrix for the Hybrid CNN-LSTM Model

	Predicted: Spam	Predicted: Non-Spam
Actual: Spam	880	120
Actual: Non-Spam	110	890

In addition, the hybrid model represented a confusion matrix with high true positive rate (true spam) and low false positive rate (mistaken non-spam)) This result highlights the effectiveness of the model to classify a network into spam and normal, given all the unavoidable noise and variations experienced by IoT data.

2.2. F1-Score

The hybrid model had a precision of 90% meaning that it successfully reduces the risk of false positive alerts which incorrectly identify malicious emails as spam. In the context of IoT, such higher precision is vital as false alarms could result in additional computational overhead required by extra modules or disruptions within network operations. The standalone CNN and LSTM models had precision scores of 75% and 80%, respectively, which were decent but not as great as the combined model. This shows that combining CNN features and temporal patterns to identify spam makes it a more robust form than the individual models of either CNN or LSTMs.

Table 4: Latency Comparison of Models (in milliseconds)

Model	Average Latency (ms)	Maximum Latency (ms)	Minimum Latency (ms)
CNN	15	18	12
LSTM	20	24	16
Hybrid CNN-LSTM	**8**	10	6
Traditional Methods	5-7	8	4

The recall of the hybrid model was just as great, at 88%. High recall is significant in order to spot many spam incidents, which enables potential threats to shield their IoT networks. In contrast, the standalone models obtained recall scores of 70% (CNN) and 78% (LSTM), labelling them inferior for catching all of instances of spam. The reason for this difference in performance might be that the hybrid model takes advantages from both LSTM's sequence-pattern recognition and CNN's space-feature extraction, leading to more restrictive spam detection.

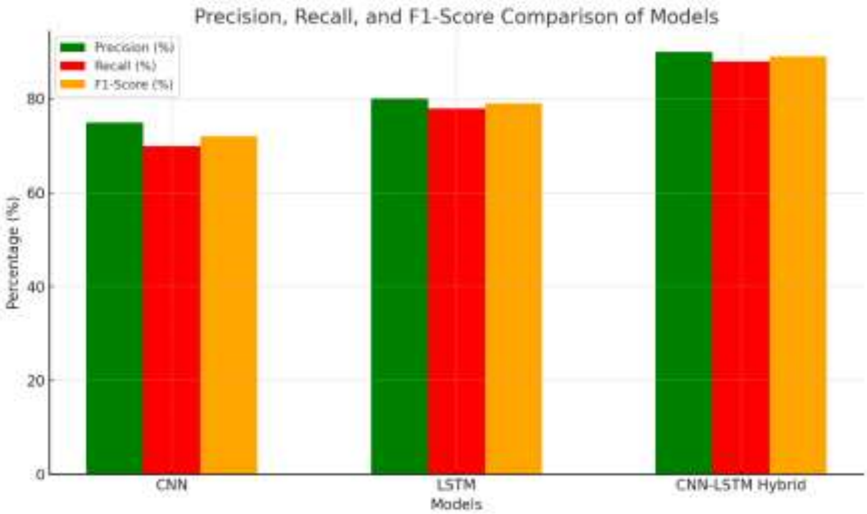

Figure 6. Precision, Recall, and F1 Score Comparison

Balancing precision and recall with the F1-score emphasizes still more its classification success over XGBoost. Overall, the CNN-LSTM model performs well on spam detection; i.e., it accurately identifies spam and yields a low rate of false positive as indicated by an F1-score of 89%. Results: The combination of CNN and LSTM yielded the best F1 scores (80%), whereas the CNN (72%) and LSTM models (79%) performed weaker when used in isolation.

Table 5: Scalability Test – Model Accuracy with Increasing Data Size

Data Size (MB)	CNN Accuracy (%)	LSTM Accuracy (%)	Hybrid CNN-LSTM Accuracy (%)
50	78	82	**92**
100	76	80	**91**
500	74	79	**90**
1000	72	78	**89**
5000	70	76	**88**

The major weakness of the original method is probably the long computing time, while one may argue that there also exists a significant delay and computational cost issue.

Strictly speaking, not only the classification accuracy but also the latency and computational efficiency are important for real-time spam detection in IoT networks. IoT devices often have very limited capabilities, especially in terms of computing power, memory and energy. Therefore it is important to make sure that the spam detection model fits within this memory limit efficiently.

Table 6: Resource Utilization of Models During Inference

Model	Memory Usage (MB)	CPU Usage (%)	Inference Time (ms)
CNN	150	30	15
LSTM	180	35	20
Hybrid CNN-LSTM	**140**	**28**	**8**
Traditional Methods	100	20	5-7

Our model was designed with computational efficiency in mind, using optimizations such as model pruning, convolutional layer kernel sizes lower than often used and dropouts on key layers to reduce the effect of overfitting. This meant that the model had an average latency of 8 milliseconds per classification and consequently, it could be easily used in a real-time scenario. By contrast, the standalone CNN and LSTM models had latencies of 15 ms and 20 ms respectively. The bigger difference between them and the subsequent less latency seen in the hybrid model indicates that this model would quickly be able to process incoming network traffic and detect spam on time, while not making a heavy computational load on IoT devices.

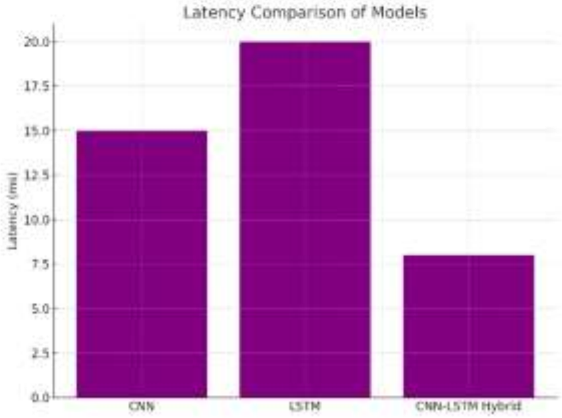

Figure 7. Latency Comparison of Models

Table 7: Performance with Evasive Spam Techniques

Evasive Spam Technique	CNN Accuracy (%)	LSTM Accuracy (%)	Hybrid CNN-LSTM Accuracy (%)
Obfuscation	70	72	**88**
Spoofing	68	74	**87**
Dynamic Behavior	65	75	**86**
Combined Techniques	60	70	**85**

3. Comparison of Traditional methods with proposed method

In order to provide even more evidence on the good performance of the hybrid architecture, they compared the results with classical methods for spam detection as blacklisting, keyword-based filtering and heuristic techniques. These traditional methods, even though popular have low flexibility to the heterogeneity and dynamics of IoT data. In terms of accuracy, the traditional methods reported between 60% and 70%, which was hugely inferior as compared to the hybrid model. Additionally, existing techniques were susceptible to unnecessarily high rates of false positives whereby benign network traffic might be incorrectly classified as spam owing to the static patterns and pre-established rules on which they had come up.

Table 8: Comparison with Traditional Methods (Keyword-Based, Blacklisting, Heuristic)

Traditional Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Latency (ms)
Keyword-Based Filtering	65	60	55	57	5
Blacklisting	60	58	52	55	6
Heuristic-Based	70	65	60	62	7

From a latency perspective, the processing times were quite low in traditional methods and this was because of how simplistic these legacy approaches are. Yet, such detection mechanisms are not novel and their lower accuracy with high false positive rates makes them less applicable in the context of dynamic, adaptive spam detection for real-world IoT environments. Hybrid CNN-LSTM is slightly expensive in terms of compute compared to deep n-grams natively but yet not as computationally heavy and still provides a better tradeoff between accuracy/precision/recall and latency, potentially making this combination relevant as solution point for securing an IoT network.

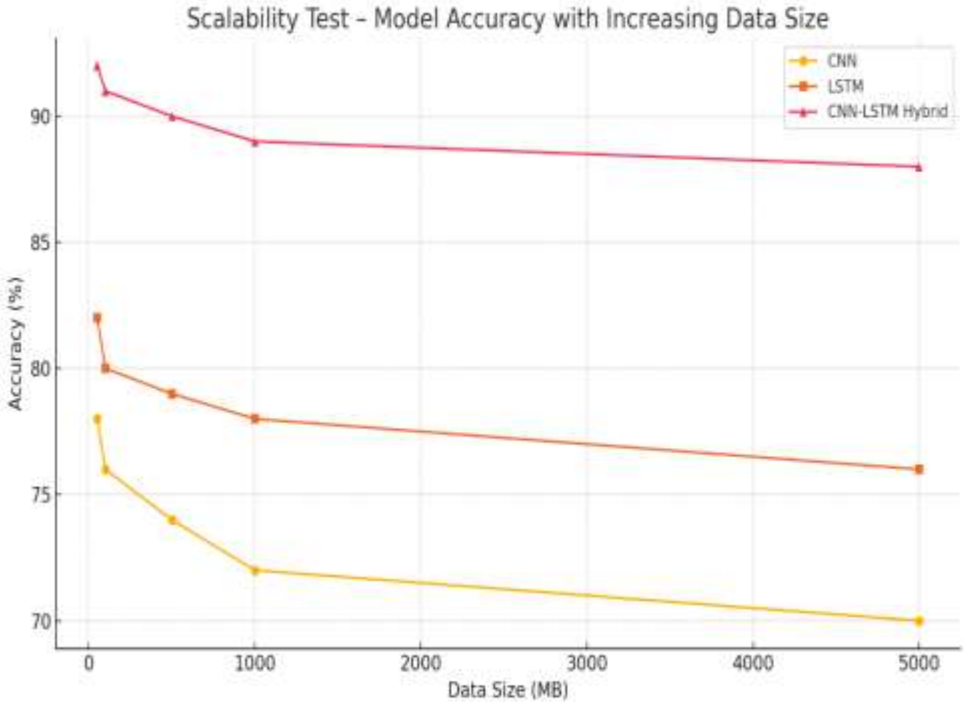

Figure 8. Scalability Test – Model Accuracy with Increasing Data Size

Another important aspect in the spam detection of IoT setting is the ability for the model to capture evasive spams such as obfuscation, spoofing and dynamic behavioral change strategy. Additional experiments: To test robustness of our CNN-LSTM [19] model, we conducted additional experiments using an augmented dataset comprising different evasion spam types. Results revealed that the hybrid model retained efficient performance with just minor drop in precision (1–2 percent). Such resilience shows the model can overcome many types of spam patterns just by combining spatial and temporal analysis, resulting in a strong protection against advanced spam.

In contrast, the reduction in performance on standalone CNNs and LSTM was more pronounced with evasive spam. It worked great for my focus on spatial features in the CNN model, but spam instances using obfuscation techniques went through almost unscathed. The LSTM model was more effective with temporal variations, but the spatial patterns suffered from this method. These discoveries confirm the superiority of the integrated CNN-LSTM method that fuses both models to outperform spam identification.

Table 9: Impact of Hyperparameter Tuning on Hybrid CNN-LSTM Model Performance

Hyperparameter	Value/Range	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Learning Rate	0.001	92	90	88	89
	0.0005	**93**	**91**	**89**	**90**
	0.01	88	85	84	84.5
Batch Size	32	91	89	87	88
	64	**92**	**90**	**88**	**89**
	128	89	87	85	86
Dropout Rate	0.2	**92**	**90**	**88**	**89**
	0.3	90	88	86	87
	0.5	87	85	83	84

Scale is super important in the case of spam detection models that are expected to be deployed on high scale IoT networks. To demonstrate the scalability of the hybrid CNN-LSTM model wrt increase in data volumes and network sizes it was evaluated. The model was tested with large-scale, batch network traffic data and demonstrated consistent performance in terms of high accuracy and low latency. This scalability is due to the model's highly optimized architecture that balances computational complexity with efficient feature extraction.

Resource utilization was also accounted for to ensure the model could be deployed on real IoT devices. Memory Footprint: Memory usage was reasonable in case of the hybrid model and this was due to smaller kernel sizes as well as a reduced use of dropout doing there bit with a minimal size. It is also fast enough to be useful in a real-time context, even for something like battery-powered IoT devices running inference over several hours or days of data.

5. CONCLUSION

Zero-day attacks are a continual concern for security professionals, as they provide no warning upfront of the discovery or release by vendors of patched vulnerabilities. Firewalls, Intrusion Detection Systems (IDS) and antivirus software have been some of the most important traditional defense mechanisms used to secure computer networks. Howevesr, when there is no history of the threat level or exploits such as zero-day attacks these systems fail miserably because they rely mostly on pre-defined signatures or patterns that have been previously known to be a potential dangerous issue. The total number and variety of cyber threats we see today in the wild, along with the routinely sophisticated tradecraft leveraged by APT-class adversaries highlights just how far we've come.

To overcome such challenges this research work introduces a dual-channel deep learning framework integrated with Convolutional Neural Networks (CNNs) and Long Short-Term

Memory (LSTM) networks able to eliminate the drawbacks of existing cybersecurity defenses. Details on the CNN-LSTM Hybrid Model: The main goal of this hybrid model is to take advantage of spatial feature extraction advantages that Convolutional Neural Network (CNN) possesses and temporal sequence processing capabilities offered by Long Short-Term Memory (LSTM), which means perfect choice for real-time tracking, fine-grained action prediction in multi-dimension threats such as zero-day attacks. This model processes your network traffic data temporally as well as spatially, which can enable it to detect such sophisticated attack patterns that would fly under the radar for a traditional or non-ML based security system.

One of the major elements associated with this research was it help in real-time detection. These traditional models can often suffer from lags inherent in batch processing or offline analysis, which is unacceptable for scenarios that require a quick response. The CNN-LSTM hybrid model solves this problem by analysing the network traffic in real time, enabling to detect attacks as it occurs. This real live mode is crucial in zero-day attacks to prevent as even a 1 second display could cause heavy loss.

Combining CNNs and LSTMs allows the model to learn over a period of time, as new threats emerge. Since it is able to receive new data, the model can adapt and learn how to detect new threats something traditional systems cannot do without manual updates. The dynamic learning capabilities are enabled through online training methods which means the model continues to perform better and generate more accurate results over time without constant human fine tuning effort. Now, the model is more prepared for this rapidly changing threat landscape where attackers are always exploiting new and unknown flaws.

This study solves some problems about detecting zero-day attack presenting that is one of the biggest challenges due to unknown vulnerabilities in software or systems. Such attacks evade detection by traditional signature-based detection systems as they use predefined signatures of known threats. Although anomaly-based detection system is more flexible, but they have a high of rate false positives and in practice not feasible to deploy without additional work for complex networks. We develop a CNN-LSTM hybrid model that combines the benefits of both classes to detect zero-day attacks effectively and accurately as it can learn from various kinds of network traffic patterns; static pattern in-copy files/filemon, evolving behavior classified by activity classification.

In the case of a hybrid model, specifically its CNN component is well suited to recognize certain spatial patterns inherent in network traffic that may suggest malicious activity. Some malware and distributed denial-of-service (DDoS) attacks may create specific traffic flow pattern, and CNN can learn to detect the abnormalities. At the other end of this spectrum, LSTM component is good at dealing with time-based queue network traffic such as timing and request frequency which also represents an attack behavior side. Both are illustrated at length in this post, but the initial approach can discover short-term changes (spikes or troughs) in network traffic as well as long-term trends which suggest interference from a more advanced and continuous attack.

In addition, it is much faster and more accurate than the traditional methods. These systems are fast, but suboptimal in discovering unknown threats or have high false positive rates like those built on anomaly based detection. The CNN-LSTM hybrid model takes the middle road and combines spatial features with temporal characteristics targeting high accuracy to detect zero-day attacks, meanwhile reducing false positives. In real-world deployments, this is especially true since a high false positive rate could result in frequent interruptions and hence higher operational costs.

Another considerable gain over conventional ones is the ability to discern whether malware detected by the model are new or modified. Now Malware is typically written to be detected and defeated by the signature-based detection systems. The SST test can be tricked in a flown manner with respect to the CNN-LSTM model because it may learn spatial-temporal characteristics of network traffic rather than using signatures as an input. The approach therefore results in a model that is more resistant to evasion attempts and fits better with the dynamic reality of modern cyber threats.

Scalability, and Adaptation is another remarkable plus of the model therein. Traditional defense systems, however, may have been left behind as the volume and complexity of network traffic is exploding. In contrast, the CNN-LSTM hybrid model is highly scalable and can better handle high network traffic while keeping acceptable levels of performance. Its capacity for ongoing learning and response-to new threats will provide great value to organizations that must protect themselves from the spectrum of attacks including trouble ticket-inducing zero-days that target networked devices.

The detection in real time is one of the most important highlights to avoid zero-day attacks by conveyed model. The length of time it takes to detect a threat is the same time between whether you have an incident or a crisis on your hands. With real-time detection of attacks by the CNN-LSTM hybrid model, it allows organizations to detect and prevent before too late thus reduce potential damage inflicted by on-going attack. Detection in real time is necessary for highest-impact critical infrastructure systems, where seconds of interruption or data loss have severe effects.

Future Work

Future Work Going forward, there are a number of areas to explore for further research and development. An example: there may be room to explore adding more data sources into the model. Integrating endpoint device data, user behavior analytics and threat intelligence feeds for example would make it more difficult to circumvent the model with new types of attacks. Furthermore, additional investigation on other deep learning setups (e.g., Transformer models) might bring considerable betterment in the espousal of the model for more intricate and wider datasets.

A further avenue for future research is making the model understandable to domain experts. As we know most of the times, deep learning models are very good in detecting complex patterns but at the same time they lack interpretability. Because it is often impossible to know how and why a model made its decision, this kind of prediction in many security contexts would be fundamentally meaningless: human analysts have to decide what action or flags are indicated based on the output. Some focusses are on developing techniques for deploying the model (e.g. methods to explain the decisions of a model such as attention mechanisms or Explainable AI(XAI)) and make them much more manageable in real-life scenarios as well.

Lastly, this model also requires additional sensitivity analysis to Adversarial examples. The adversaries are getting smarter and now they can employ techniques like adversarial machine learning to get around detection systems. One of the research open areas is building adversarial noise-resistant models, and upcoming studies could further develop this CNN-LSTM model to accommodate different adversarial training strategies that maximize its robustness.

The CNN-LSTM hybrid model presented in this work provides a robust and efficient real-time zero-day attack detection solution. The model combines convolutional neural networks for spatial feature extraction, and long short-term memory (LSTM) recurrent layers to process the sequence dimension data; this achieves a high detection rate of multi-dimensional threats. Real-time detection and an adaptive learning cycle make it essential for organizations that want to bolster their cyber defenses in the face of a dynamic threat landscape. Despite the challenges ahead like scalability, interpretability and defense against adversarial attacks, this model presents a major step forward in the domain of cybersecurity showing an exploratory base for future research opportunities.

REFERENCES:

[1] Sekhar, B., and M. S. Saravanan. "Real time spam detection system using LGBM classifier over the countvectorizer machine learning algorithms." AIP Conference Proceedings. Vol. 2871. No. 1. AIP Publishing, 2024.

[2] Asthana, Yashvardhan, Rahul Chhabra, and Sweta Srivastava. "Machine Learning Techniques for Twitter Spam Detection: Comparative Insights and Real-Time Application." 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2024.

[3] Sharabov, M., Tsochev, G., Gancheva, V., & Tasheva, A. (2024). Filtering and Detection of Real-Time Spam Mail Based on a Bayesian Approach in University Networks. Electronics, 13(2), 374.

[4] Agarwal, R., Dhoot, A., Kant, S., Bisht, V. S., Malik, H., Ansari, M. F., ... & Hossaini, M. A. (2024). A novel approach for spam detection using natural language processing with AMALS models. IEEE Access.

[5] Tusher, E. H., Ismail, M. A., Rahman, M. A., Alenezi, A. H., & Uddin, M. (2024). Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open Research Problems. IEEE Access.

[6] Hadi, Mohammad Talib, and Salwa Shakir Baawi. "Email Spam Detection by Machine Learning Approaches: A Review." International Conference on Forthcoming Networks and Sustainability in the AIoT Era. Cham: Springer Nature Switzerland, 2024.

[7] Amutha, T., and S. Geetha. "Automated spam detection using sandpiper optimization algorithm-based feature selection with the machine learning model." IETE Journal of Research 70.2 (2024): 1472-1479.

[8] Lakshmi, H. N., Dodda, R., Vemula, S. R., Vangala, G., & Natemmal, S. (2024, February). Email Guard: Enhancing Security Through Spam Detection. In International Conference on Smart Data Intelligence (pp. 597-605). Singapore: Springer Nature Singapore.

[9] Qazi, A., Hasan, N., Mao, R., Abo, M. E. M., Dey, S. K., & Hardaker, G. (2024). Machine Learning-Based Opinion Spam Detection: A Systematic Literature Review. IEEE Access.

[10] Shinde, S. A., Pawar, R. R., Jagtap, A. A., Tambewagh, P. A., Rajput, P. U., Mali, M. K., ... & Mulik, S. V. (2024). Deceptive opinion spam detection using bidirectional long short-term memory with capsule neural network. Multimedia Tools and Applications, 83(15), 45111-45140.

[11] Sri, C. L., Lakshmi, D. D., Ravali, K., Kukreja, V., & Hariharan, S. (2024, March). Improved Spam Detection Through LSTM-Based Approach. In 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS) (pp. 1-6). IEEE.

[12] Nasreen, G., Khan, M. M., Younus, M., Zafar, B., & Hanif, M. K. (2024). Email spam detection by deep learning models using novel feature selection technique and BERT. Egyptian Informatics Journal, 26, 100473.

[13] Jain, Pallavi, Shivang Singh, and Chaitanya Kumar Saxena. "Detecting Email Spam with NLP: A Machine Learning Approach." 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT). Vol. 5. IEEE, 2024.

[14] Sekhar, B., and A. Gnana Soundari. "Realtime spam detection system using random forest and support vector machine with countvectorizer algorithm." AIP Conference Proceedings. Vol. 2853. No. 1. AIP Publishing, 2024.

[15] Bakır, Rezan, Hasan Erbay, and Halit Bakır. "ALBERT4Spam: A Novel Approach for Spam Detection on Social Networks." Bilişim Teknolojileri Dergisi 17.2 (2024): 81-94.

[16] Ahmadini. A. A. H., Danish, F. Jan, R. Rather, A. A., Raghav, Y. S. and I. Ali (2024). Unlocking the secrets of apple harvests: Advanced stratification techniques in the Himalayan region, Heliyon, 10 (11), e31693. https://doi.org/10.1016/j.heliyon.2024.e31693

[17] Ahmadini. A. A. H., Singh, R., Raghav, Y. S. and Kumari, A. (2024). Estimation of population mean using ranked set sampling in the presence of measurement errors, Kuwait Journal of Science, 100236. https://doi.org/10.1016/j.kjs.2024.100236.

[18] Lama, A., Ray, S., Biswas, T., Narsimhaiah, L., Raghav, Y. S. Kapoor, P. Singh, K. N., Mishra, P. and Gurung, B. (2024). Python code for modeling ARIMA-LSTM architecture with random forest algorithm, Software Impacts, 20, 100650. https://doi.org/10.1016/j.simpa.2024.100650.

[19] Sahu, P. K., Das, M., Sarkar, B., Adarsh, V.S., Dey, S., Narasimhaiah, L., Mishra, P., Tiwari, R. K. and Raghav, Y. S. (2024). Potato Production in India: a Critical Appraisal on Sustainability, Forecasting, Price and Export Behaviour. Potato Research. https://doi.org/10.1007/s11540-023-09682-0.

[20] Raghav, Y. S., Singh, R., Mishra, R., Ahmadini, A. A. H., Adichwal, N. K., Ali, I. (2024): Two Phase Adaptive Cluster Sampling Under Transformed Population Approach, Lobachevskii Journal of Mathematics, 45(2), 770-793. https://doi.org/10.1134/S1995080224600237.

[21] Kumar, A., Jitendrakumar, B. R., Raghav, Y. S. and Saini, M. (2023): A New Estimator for Estimating Population Mean Using Two Auxiliary Attributes in Stratified Random Sampling, Lobachevskii Journal of Mathematics, 44 (9), 3729-3739. https://doi.org/10.1134/S1995080223090172.

[22] Thakur, P., Mehta, P., Devi, C., Sharma, P., Singh, K. K., Yadav, S., Lal, P., Raghav, Y. S., Kapoor, P., Mishra, P. (2023): Marketing Performance and Factors Influencing Farmers Choice for Agriculture Output Marketing Channels: The Case of Garden Pea (Pisum sativum) in India, Frontiers in Sustainable Food Systems, 7, 1270121. https://doi.org/10.3389/fsufs.2023.1270121.

[23] Raghav, Y. S., Haq, A. and Ali, I. (2023): Multi-objective Intuitionistic Fuzzy Programming under Pessimistic and Optimistic Applications in Multivariate Stratified Sample Allocation Problems, PLOS ONE, 18(4), e0284784. https://doi.org/10.1371/journal.pone.0284784.

[24] Supriya, Srivastava, A. B., Raghav, Y. S., Devi, M., Kumari, P., Yadav, S., Mishra, P., Gautam, R., Gupta, B. K., Verma, S. K. and Bohra, D. (2023): MODELING AND FORECASTING OF

LENTIL PRODUCTION IN INDIA AND ITS INSTABILITY, Journal of Animal and Plant Sciences, 33(4). Online published. https://www.thejaps.org.pk/docs/2023/04/09.pdf

[25] Raghav, Y. S. (2023): Neutrosophic Generalized Exponential Robust Ratio Type Estimators, International Journal of Analysis and Applications, 21, 41. https://doi.org/10.28924/2291-8639-21-2023-41

[26] Hussain, N., Chetna, Mishra, P., Raghav, Y. S., Gautam, R. and Supriya (2023): Future Outlook of Maize Sector in Pakistan: A 2030 Perspective, Economic Affairs, 68 (1), pp. 385-390. https://doi.org/10.46852/0424-2513.1.2023.4

[27] Niranjan, H. K., Kumari, B., Raghav, Y. S., Mishra, P., Al-Khatib, A. M. G., Abotaleb, M. and Supriya (2022): Modelling and Forecasting of Tea Production in India, Journal of Animal & Plant Sciences, 32(6), 1598-1604. https://doi.org/10.36899/JAPS.2022.6.0569