Revolutionary Advances in Computing and Electronics: An International Journal
Volume.1, Issue.1, pp. 01-16, July-Sept 2025

Received: 09/05/2025, Accepted: 25/05/2025

Research Article

CNN-LSTM Powered Network IDS for Adaptive Cyber Defence

Md Aadil Hasan, Dev Sharma
School of Computer Science Engineering and Technology, Bennett University, Greater Noida, UP, India
Email: aadilhasan1185@gmail.com, devsharma241995@gmail.com

Abstract

A Network IDS (NIDS) is devised to examine network traffic to detect signs of malicious behaviour or violations of protection policies. It is an essential piece of equipment in the fight to improve cybersecurity via early threat detection and response. Existing machine learning approaches, though efficient, are generally bogged down by significant manual feature engineering, which restricts their flexibility to adapt to dynamic attack scenarios. Deep learning approaches, with the ability to automatically learn high-level features, provide a robust alternative to designing effective IDS. This work proposes a novel hybrid deep learning architecture to synergistically integrate CNN and LSTM networks for tackling the complexity of network intrusion detection. The CNN module performs well in detecting spatial patterns of network traffic data, and the LSTM module incorporates temporal dependencies to facilitate exhaustive analysis of sequential attack patterns. To improve model efficiency and avoid overfitting, batch normalization and dropout layers are strategically integrated in the architecture. The model is extensively tested on three diverse datasets, CIC-IDS2017, UNSW-NB15, and NSL-KDD, covering a broad range of contemporary attack types. Experiments are performed for binary and multiclass classification tasks, and performance metrics are evaluated based on a confusion matrix. Key performance metrics, like false alarm rate, accuracy, F1 score and detection rate, define the model's performance in intrusion detection with high accuracy while avoiding a high false positive rate. The outcome proves the model's robust performance across diverse network environments, varying from wired to wireless networks, and its applicability in detecting known and novel threats. By tapping the power of automated feature extraction and sophisticated neural network design, the work critically contributes to a scalable and efficient solution to existing network security, opening the door to real-time, adaptive intrusion detection across complex digital terrain.

Keywords: IDS, DL, CNN, LSTM, Hybrid model, False Alarm rate, Datasets, Cyber threats

1 Introduction

In today's world, Network Intrusion Detection Systems (NIDS) are super important. Think of them as the watchdogs of your network, constantly monitoring traffic to sniff out any unauthorized access, cyber-attacks, or policy slip ups. They're basically what keeps your digital stuff safe from real-world threats. And with all the new tech like IoT devices, big data, and cloud computing and how much it depends on everything being connected, network security has become a bigger deal than ever. Even a tiny weak spot in your network can let someone compromise the whole thing [1]. The old ways of protecting ourselves, like encryption and firewalls, aren't enough anymore to stop today's clever cybercriminals [2]. That's why cybersecurity folks focus on making smarter, more adaptable Intrusion Detection Systems (IDS). The goal To keep the data private, make sure it's not messed with, and ensure it's always available. A good IDS needs to spot known and brand-new threats and do it accurately without false alarms [3] [4]. Figure 1 represents the Global usage of NIDS in 2025, demonstrating rising deployment levels as a strategic defence against evolving cyber threats. The data emphasizes how organizations prioritise NIDS for enhanced real-time network protection.

*Corresponding author: Md Aadil Hasan, SCSET, Bennett University (aadilhasan1185@gmail.com)

Now, there are generally two main ways these IDS systems work: signature-based detection and anomaly detection, also known as misuse detection, which is all about looking for patterns of known threats.It's usually accurate and doesn't give you a lot of false alerts. However, it's not excellent at spotting new attacks it hasn't seen before [5]. Anomaly detection is good at finding unknown threats by noticing when things aren't behaving normally. The catch is that it can sometimes give you more false positives. Attacks are getting more varied and unpredictable; detecting anomalies is crucial for keeping networks secure.

This is where Intelligence algorithms come into play, where these systems can learn from data without needing

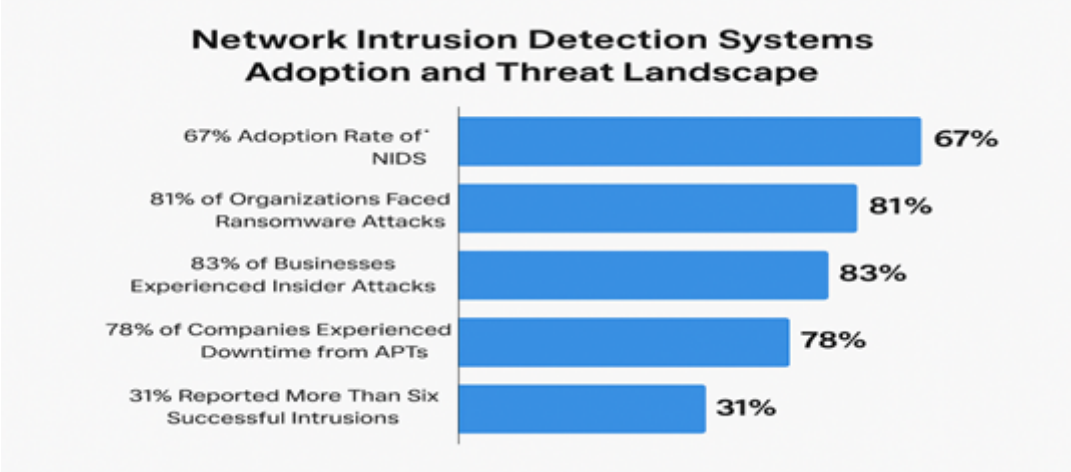

Figure 1: Global Adoption Rate of NIDS – 2025

someone to constantly watch them, which means intrusion detection becomes more automated and adaptable. ML and Deep learning are two key players here. ML models rely on manually created features to classify and detect network traffic. In contrast, DL models, powered by neural networks, can automatically learn these features, often leading to better results [6].

Over the past years, many different intrusion detection models have used AI techniques. However, many still struggle with accuracy, have trouble detecting threats, and give too many false alarms when put to work in complex, real-world situations. The fact that networks are so diverse and constantly changing makes it even harder for traditional IDS solutions to be effective. To tackle these problems, this paper introduces a hybrid model of deep learning that combines CNN & LSTM networks in which the CNN layer is designed to pull out spatial features from network traffic. In contrast, the LSTM layer detects temporal dependencies, allowing the model to analyse the data's static and dynamic aspects. The hope is that this combination will improve accuracy and reduce false positives using the best of both architectures.

The hybrid CNN+LSTM model is trained using three well-known datasets: WSN-DS, CIC-IDS, and UNSW-NB15. The evaluation shows that the standalone CNN, LSTM, traditional machine learning models, and the hybrid model came out on top, with higher accuracy and better threat detection rates. This makes it a promising option for defending against today's advanced network attacks.

The paper is outlined as follows: Section II summarises machine learning algorithms for detecting network intrusions. Meanwhile, the increasing usage of deep learning strategies is discussed herein, and this hybrid model indicates that it can solve the problem at hand. Section III describes the architecture of the CNN & LSTM hybrid model using data from supervised machine learning experiments. Each parameter in the model was fully expressed, and the preprocessing techniques were introduced. Data used for training and evaluating the model are presented in Section IV. The datasets were carefully chosen to represent various characteristics of network attributes. Section V describes the experimental setup of the proposed hybrid model and defines performance metrics used to assess the model. The model's effectiveness is assessed using various performance metrics, and the outcomes are concisely summarized. The results and suggestions for further research are also included at the end of the paper.

2 Types of IDS

Security professionals are looking to develop models capable of identifying known and unknown attacks in the network, aiming to prevent any potential harm to network systems. As will be discussed next, the techniques used to build IDS (Intrusion Detection Systems) are categorized into machine and deep learning frameworks.

2.1 IDS using ML

Machine Learning (ML) has always been a key component of intrusion detection systems. ML techniques apply supervised learning algorithms such as SVM, Decision tree and Naïve Bayes, and unsupervised techniques like Self-Organizing Maps and k-means clustering [7]. The primary purpose of these algorithms is to boost the system's ability to detect threats. Using trained datasets, ML algorithms are employed to identify attacks and anomalies. These algorithms typically address problems related to regression, classification, and clustering. Earlier research mainly relied on datasets like NSL-KDD, KDD-CUP99 and DARPA. Although some models achieved reasonable outcomes, these datasets are outdated and only cover basic types of attacks [8]. In today's rapidly evolving network environment, creating an efficient IDS requires extensive and updated datasets; thus, relying solely on traditional ML models that work well with smaller datasets will not be sufficient.

2.2 IDS using DL

Deep learning algorithms are a segment of machine learning techniques that use neural networks with several hidden layers. It can also process unstructured and unlabelled data, not just structured inputs [9]. Deep learning offers various performance advantages that make it well-suited for IDS development, including the robustness and scalability of its algorithms and the ability to manage distinct forms of data [10]. These algorithms are designed to solve sophisticated problems like machine translation, pattern identification and search engine optimization [11]. Unsupervised neural models, including Autoencoders, RBMs, and Deep Belief Networks, are usually applied for feature extraction [12]. Multi-layer perceptrons are also used in various fields to reduce error rates while training [13]. Some of the most popular algorithms of deep learning are CNN and RNN. CNN is especially effective at auto discovering spatial features without requiring hand engineered feature design, to curb overfitting by minimising the no. of trainable parameters and promoting better generalization [4]& at the same time, RNNs are most used in the fields of NLP, speech recognition, and video processing due to their ability to recognize sequential data patterns. In addition, LSTM networks were re-discovered to overcome the vanishing gradient problem experienced by RNNs during training.

3 Proposed Methodology

This study has an intrusion detection system with CNN & LSTM layers. The proposed methodology is shown in Fig. 2, which explains the whole experiment implementation from data collection to outcome evaluation.

Table 1: Test cases

Dataset	**Binary Classes**	**Binary Count**	**Multiclass Types**	**Multiclass Count**
CIC-IDS 2017	Malicious, Normal	2	Web Attack, FTTP, Patator, SSH-Patator, PortScan, Normal	6
UNSW-NB15	Malicious, Normal	2	Fuzzers, Worms, Generic, Exploits, Normal, Analysis, Backdoor, Shellcode	10
WSN-DS	Malicious, Normal	2	Regular, Flooding, TDMA, Grayhole, Blackhole	5

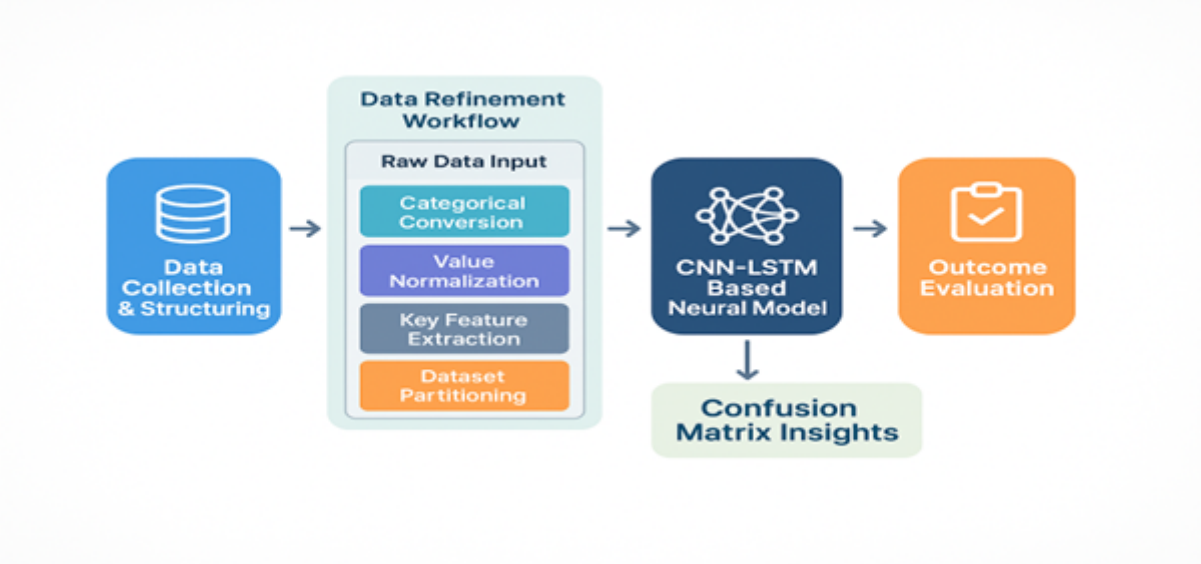

Figure 2: Proposed Methodology

3.1 Data Collection and Structuring

The 1st vital step in designing a good IDS is the selection of a proper dataset. The dataset must include normal and malicious traffic to simulate the real scenarios the model will encounter. This research employs WSN-DS, CIC-IDS, & UNSW-NB15 datasets, which are publicly available and relatively newer. All three datasets include normal and malicious traffic data that have been recently updated and consist of less redundant data. These datasets provide realistic, recent, and diverse network traffic for practical IDS training.

- **CIC-IDS2017**: In 2017, the Canadian Institute for Cybersecurity 2017 released the CIC-IDS2017 dataset, which records eleven attack categories—Brute Force, Port Scanning, DoS, XSS, SSH-Patator, FTP-Patator, and SQL injection. It uses 80 flow attributes to characterize each instance and is widely adopted for modern cyber threat analysis

- **UNSW-NB15**: It was developed by the Australian Centre for Cyber Security in 2015. UNSW-NB15 combines benign traffic with nine malicious classes, such as Backdoor, DoS, Fuzzers, Analysis and Exploits. The traffic was obtained from real-world sources like Symantec's BID, Microsoft's MSD, and CVE databases, which provide complete coverage of vulnerabilities and attack types.

- **WSN-DS**: Proposed in 2016, WSN-DS targets wireless sensor networks employing the LEACH routing protocol. [14] It provides 23 extracted features and logs regular activity and four DoS types—TDMA, Grayhole, flooding, and Blackhole—to support sensor network intrusion detection research

3.2 Data Refinement Workflow

- **Raw data Input**: The data used in this research paper were open access. Data were converted from PCAP files and stored in CSV format. In this step, the Pandas library was employed to read every dataset. The data were cleaned by deleting null values and duplicate entries after reading them, then reading them for successive processes. Categorical conversion Datasets were encoded here to transform label values into numeric values, which need to be processed by neural networks. As the labels were categorical, the One Hot Encoder was used to express the benign and malicious traffic labels in their numeric representations.

- **Value Normalisation:** Normalization was used to scale the datasets and improve the within-range features. Standard deviation and mean value variations would otherwise affect the efficiency of this model during training. The data is normalised using the StandardScaler from sklearn. Preprocessing: Adjust the data so that the

mean is 0 and the standard deviations are 1. The Standard Scaler was done using the sklearn—preprocessing library.

- **Key Feature Extraction:** Feature reduction, or feature selection, is crucial in minimizing the feature set based on specified criteria. It speeds up model construction and reduces the training computational cost, improving performance overall. SelectKBest in sklearn. Feature selection was employed in this work. SelectKBest identifies and selects features with the top scores. The function returns a list of feature names and scores, and the best features were chosen based on a given K value.

- **Dataset partitioning:** The data were initially partitioned so that 80% served as the training pool and 20% was held out for testing. The training portion was then split again into a smaller and a validation segment for hyperparameter tuning. Stratified K-fold validation was performed with the specified number of splits to maintain class balance and identify the best division.

3.3 Integrated Deep Learning Model

CNN is designed to extract spatial features, whereas LSTM focuses on learning temporal relations. Leveraging the ability of the CNN to extract higher-level features from substantial datasets, this model begins with CNN layers. Initially, the data passes through convolutional layers where filters identify essential features to construct a feature map. Now the map undergoes maximum pooling to retain dominant features, subsequently the next step is batch normalization technique.
The resulting output is forwarded to an LSTM layer that identifies temporal characteristics. A dropout layer is subsequently applied to prevent overfitting. This sequence of (CNN)&(LSTM) layers is repeated 3 times with varying neuron and filter configurations, ending with a completely connected dense layer employing the SoftMax activation function, which produces the final classification outputs. Figure 3 illustrates the proposed deep learning architecture. The model architecture consists of repeated blocks, each containing convolution and pooling layers, LSTM, dropout, batch normalization and fully connected layers, repeated three times.

3.3.1 Convolutional Neural Network

It consists of alternating convolution, activation, and pooling stages that together learn to recognize spatial hierarchies in data such as images. During convolution, a small filter K of size $M \times N$ is slid over the input map I, computing at each position (r, s) the sum of element-wise products plus a bias c. Here, $I(r+m, s+n)$ denotes the input pixel at offset (m, n), $K_{m,n}$ is the corresponding learnable weight, c is a scalar bias added to every location, and $\sigma(\cdot)$ functions as a nonlinear activator, like ReLU. The resulting feature map Y passes through a pooling layer—often maxpooling—which reduces each patch to its maximum value, shrinking spatial dimensions, limiting overfitting, and providing modest translation invariance. By stacking many such layers, a CNN automatically progresses from detecting low-level edges and textures in initial layers to identifying intricate patterns and objects in deeper layers.

$$Y(r,s) = \sigma\left(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} K_{m,n} \cdot I(r+m, s+n) + c\right) \tag{1}$$

3.3.2 Batch Normalization Method

It is a method used to mitigate the internal covariate problem, which shifts by ensuring that inputs to each layer maintain a consistent distribution throughout training. For every mini batch, activations are first centred by subtracting the batch mean and then scaled to unit variance, after which they are optionally rescaled and shifted using learnable parameters. This process not only smooths the loss surface, permitting larger learning rates and faster convergence, but also provides a regularising effect that can reduce overfitting. In practice, batch normalization is inserted between the linear (or convolutional) transformation and the nonlinear activation, stabilising gradient flow in deep architectures.

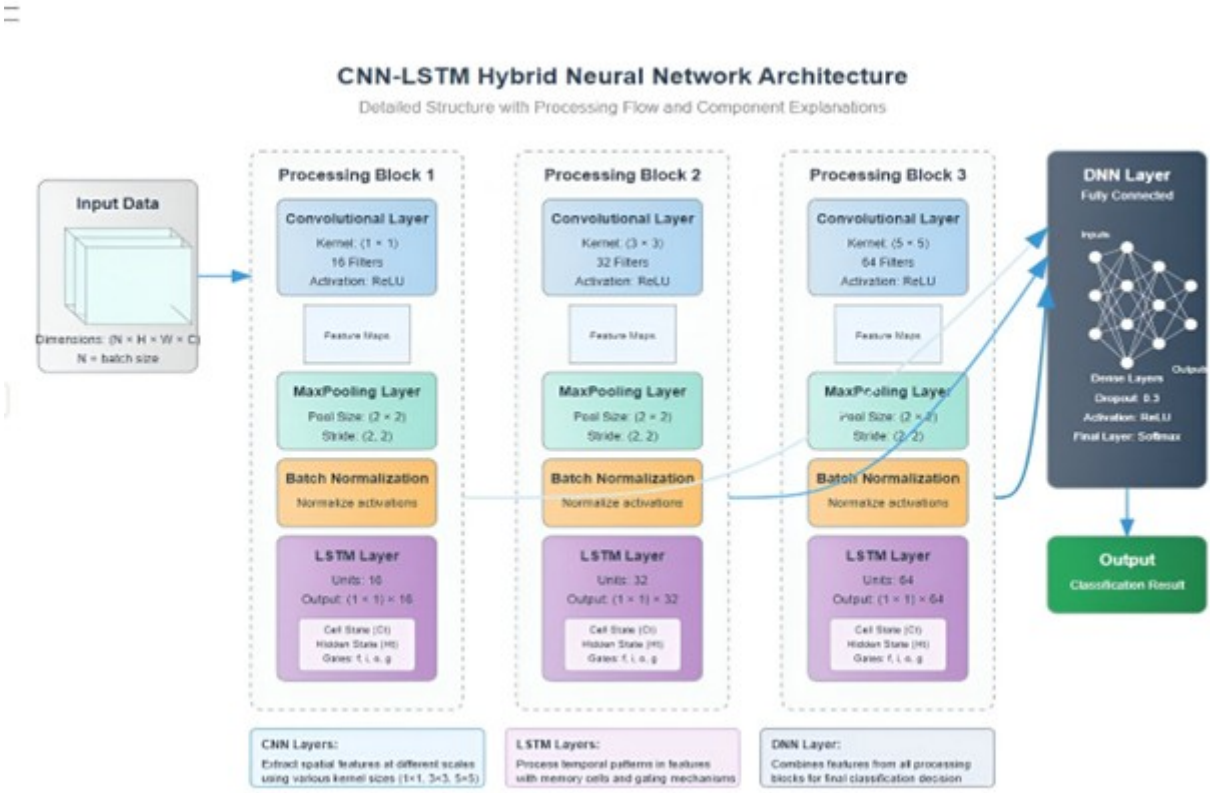

Figure 3: CNN-LSTM Layered Structure

Moreover, if advantageous, the learned shift (β) and scale (γ) parameters allow the network to recover the original representations, preserving its expressive power. Batch normalization enhances training efficiency and model robustness with minimal computational overhead

$$Y^{(i)} = \gamma \left(\frac{X^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta \tag{2}$$

3.3.3 Long-Short Term Memory

LSTM (Long Short-Term Memory) networks consist of memory cells and specially designed gates that determine what information flows in and out of the network. There are four fundamental LSTM units: the **forget gate**, **tanh gate**, **input gate**, and **output gate**, which act in tandem to decide what information to retain and what to discard so that the network can learn temporal patterns in the data.

In an LSTM, inputs and outputs are vectors with the same dimension as the underlying memory cell; such vectors are denoted as $X(t)$. The forget gate takes into account the previous hidden state $h(t-1)$ and the current input $X(t)$ to decide which information from the past should be forgotten or retained. The input gate then determines which new information from $X(t)$ should be added to the memory cell.

The cell state $C(t)$ is updated by combining the outputs of both the forget and input gates, using the $\tanh$ activation function to regulate the magnitude of the update. Finally, the output gate uses a sigmoid activation to determine which components of the updated cell state should contribute to the new hidden state $h(t+1)$. This hidden state is also modulated by a $\tanh$ function applied to the updated cell state. As a result, the LSTM network is capable of efficiently handling both short-term and long-term dependencies in sequential data.

$$O(t) = \sigma \left(b + U \times X(t) + W \times h(t-1) \right) \tag{3}$$

Table 2: Confusion Matrix – Describes the conventional confusion matrix employed for assessment

	Positive Estimation	**Negative Estimation**
Observed Positive	(A) True Positive	(C) False Negative
Observed Negative	(B) False Positive	(D) True Negative

3.3.4 Dropout

During training, neurons are randomly deactivated at every epoch using the dropout technique [7]. This method prevents overfitting in deep neural networks, where the model might otherwise memorize the trained data instead of generalizing to new inputs. This study incorporated a dropout layer with a dropout rate of 0.2.

3.3.5 Completely Connected Layer

The output layer processes the extracted feature maps. A completely connected layer links every neuron to all neurons in the previous layer, and it handles classification tasks and utilizes the SoftMax activation function to generate output probabilities. It converts the processed data into a one-dimensional vector, assigns the inputs to the correct class, and generates the final output.

3.3.6 Evaluation

The Intrusion Detection System (IDS) performance evaluation relies on confusion matrix metrics presented in Table 2. Normal traffic data leads to misclassification as dangerous behaviour, corresponding to a True Positive (A) in this environment. A False Positive (B) represents benign traffic, even though it gets classified wrongly as an attack. The IDS correctly labels malicious traffic when it detects it as True Negative (D), but it incorrectly detects normal traffic as False Negative (C). The values present in the confusion matrix result in four evaluative metrics, which combine Detection Rate (DR) and Accuracy (AC), as well as Precision (P) and the False Alarm Rate (FAR). The accuracy metric calculates the correct identification rate of all recorded data items. The detection rate signifies how well systems detect genuine attack records. The detection metrics for system accuracy are Precision (Pr), which represents the capacity to prevent false classification of normal traffic as malicious. The False Alarm Rate (FAR) describes how many normal sessions are wrongly marked as attacks.

$$\text{AC} = \frac{A+D}{A+D+B+C}, \quad \text{DR} = \frac{A}{A+C}, \quad \text{FAR} = \frac{B}{B+D}, \quad \text{P} = \frac{A}{A+B} \tag{3}$$

Table 3: Performance accuracy of various learning algorithms applied to binary classification using different datasets

Layers	**Model**	**CIC-IDS2017**	**UNSW-NB15**	**WSN-DS**
1	CNN	97.53	93.11	99.54
1	LSTM	98.89	93.19	99.63
1	CNN-LSTM	98.51	93.63	99.59
2	CNN	99.26	93.20	99.35
2	LSTM	99.39	93.37	99.35
2	CNN-LSTM	99.60	93.76	99.54
3	CNN	98.96	93.66	99.61
3	LSTM	99.21	93.35	99.61
3	CNN-LSTM	**99.61**	**94.69**	**99.62**

4 Experimental Setup and Performance

The standardized hardware selection for model testing included a Dell XPS 15 with an Intel(R) Core (TM) i9-13900H processor 5.40 GHz speed, combined with 32.00 GB system memory. This model development relied on TensorFlow as the deep learning framework, with Pandas and Keras libraries commonplace in this field.
The evaluation encompassed both multiclass and binary classification tasks. The binary classification scenario categorised Data instances as ”benign” or a specific ”attack” class. The multiclass classification extended this to differentiate among several distinct attack categories.

4.1 Different Learning Algorithms Comparisons

The research is concentrated on finding the best model architecture at the beginning. The goal demanded a performance benchmark involving independent CNNs and LSTMs, and also tested hybrid models that combined LSTM elements with CNN components (mainly CNN-LSTM systems). The outcome of this model comparison evaluation appears in Table 3. Looking at the CIC-IDS binary dataset, the CNN-LSTM architecture with 3 layers achieved the highest accuracy, reaching 99.61% and the 2-layer CNN-LSTM structures at 99.60The results for the UNSW-NB binary classification. Again, the three-layer CNN-LSTM model performed best, achieving an accuracy of 94.69%, and the two-layer CNN-LSTM models reached 93.76%.
The WSN-DS dataset showed a slightly different trend. As shown in Table 3, CNN-LSTM configurations with three layers and one layer also performed strongly, achieving 99.62% and 99.59% accuracy, respectively. After this initial comparison of the three learning algorithms, the remainder of the research was focused on the CNN-LSTM hybrid structure, given its consistently strong performance across multiple datasets.

4.2 Selected Features on CNN-LSTM

In the second phase, the focus shifted to selecting the most effective features for this model. The process began by experimenting with different feature subsets using the CIC-IDS dataset and a one-layer CNN+LSTM architecture. Specifically, 5 experiments were run, varying the number of features at 24, 40, 50, 60, and 78. For the UNSW-NB15 dataset, 3 experiments were conducted using 24, 32, and 42 features. Lastly, the model's performance was assessed on the WSN-DS dataset using feature sets comprising six, twelve, and eighteen attributes. The outcomes of these evaluations are provided in the subsequent tables. Feature selection was performed using SelectKBest, which ranks features based on their scores, and the features with the highest scores were chosen. The results obtained using the binary CIC-IDS with 24 features achieved an accuracy of 97.33% and a detection rate of 99%. Increasing the number of features to 40 improved the accuracy to 99.5% and the detection rate to 99.4%. With 50 features, we got the best results with an accuracy of 99.60% and a 99.55% detection rate. With 60 features, it got an accuracy of 99.7% and a 99% detection rate, and 78 features got an accuracy of 99.57% and a 99.53% detection rate. Based on these initial findings, 60 features give the maximum accuracy. However, 50 features gave the best detection rate, the maximum F1-score and the minimum false alarm rate. Consequently, the experiment continues using fifty features. For binary classification results on the UNSW.NB15, all 42 available features were used. First, with a single-layer CNN+LSTM, 24 features yielded 93.58% accuracy and 94.6% detection rate. With thirty-two features, the accuracy and detection rate improved to 93.70% and 94.81%, respectively. The best results were achieved using all 42 features, with an accuracy of 93.8% and a 94.85% detection rate. Notably, the lowest FAR was observed when using 42 features. Furthermore, training the model with 42 features required less time than training with 32 features. Therefore, further testing was conducted using all 42 features from the UNSW.NB15 dataset. In the feature selection process using the binary WSN-DS dataset, a model using 18 features performed optimally. The proposed configuration demonstrated strong performance, achieving 99.59% accuracy and a 98.28% detection rate. For comparison, models using 12 and 6 features achieved accuracy scores of 98.12% and 97.61%, with corresponding detection rates of 88.90% and 97.05%, respectively. Furthermore, the IDS model was also trained using the complete feature set of the WSN-DS dataset.
The evaluation of different optimization algorithms, which is compared with Adam optimizer, produced the results mentioned above, with a model based on RMSprop. When applying RMSprop to a one-layer CNN-LSTM model, the following accuracy and detection rates were observed: 99.53% and 99% for the CIC-IDS dataset, 93.58% and

93% for the UNSW.NB15 dataset, whereas 99.61% and 98.28% for the WSN-DS dataset [4]. Given the consistently higher accuracy and detection rates achieved with the Adam optimizer, it was selected for subsequent experiments.

4.3 Evaluating CNN-LSTM Architectures: Layer Configurations and Hyperparameters

The impact of different layer arrangements, neuron counts, fully connected (FC) layers, and dropout rates on model performance was investigated. For the CIC-IDS dataset, the best testing accuracy (99.61%) was achieved by a three-layer CNN-LSTM model using a dropout rate of 0.2 and one FC layer. Close behind were a two-layer model (dropout rate of 0.2 and two FC layers) with an accuracy of 99.56%, and a single-layer model (dropout rate of 0.2 and two FC layers) with an accuracy of 99.57%.
Analyzing validation accuracy, loss, and False Acceptance Rate (FAR) more closely, the three-layer configuration emerged as the best, yielding identical testing and validation accuracy (99.61%) and the lowest FAR (0.11).
For the UNSW-NB15 binary dataset, the single-layer model with a dropout rate of 0.2 and one FC layer achieved the highest testing accuracy (93.72%). However, the three-layer CNN-LSTM model, despite having a slightly lower testing accuracy, outperformed in validation accuracy (93.8%), showed the lowest loss (11), and had the smallest FAR (6.2), making it the more reliable choice overall.
In experiments using the WSN-DS dataset, the highest testing accuracy was achieved by a single-layer CNN+LSTM model with a dropout rate of 0.5 and two FC layers. Nonetheless, when prioritizing validation accuracy (99.61%), loss, and FAR (0.90), the three-layer configuration again proved superior, demonstrating its value even without the top initial testing accuracy.
Across all datasets, the three-layer CNN-LSTM architecture [15] consistently provided a strong balance of high accuracy, low loss, and low FAR, indicating that it can serve as a dependable model choice for various intrusion detection tasks [16].

4.4 CNN-LSTM model utilizing cross-validation on Stratified K-Fold

After settling on the best setup for the model, including the number of layers, neurons, rate of dropout, and fully connected layers, the next step involved tuning the Stratified K-Fold validation parameter. Let's dive into the findings.
CIC-IDS Results: Table 4 breaks down the performance metrics for the CIC-IDS dataset, looking at both binary (normal vs. attack) and multiclass (specific attack types) classification. It hits a peak accuracy of 99.65% at both K=8 and K=4. Regarding snagging those attacks, it gives a top 99.71% detection rate for binary & a whopping 99.96% for multiclass, both at K=8. The minimum false alarm rate (FAR) was 0.1, achieved at K=10 for multiclass & K=8 for binary. Precision values and F1 score are also included in the table for a complete picture.

- **UNSW-NB15 Results:** Jumping over to the UNSW-NB15 dataset, Table 4 highlights a clear difference in performance between binary and multiclass classification. The best binary accuracy, 93.96%, showed up at K=6, while multiclass peaked at 82.3% at K=4. Detection rates followed suit, with K=8 giving us the best numbers: 94.54% for binary and 82.42% for multiclass. The lowest FAR here was 2.2 at K=4 for multiclass and K=8 for binary.

- The WSN-DS simulation provides the following results. Table 4 demonstrates deviation in K's performance level. Model performance reached maximum accuracy levels in binary (99.68%) and multiclass (99.44%) operations when K reached 10. The K=10 level achieved 98.15% accuracy in binary classification & 98.84% accuracy in the multiclass task. Raw values for the lowest FAR measurements reached 0.11 at K=6 in binary identification and 0.67 at K=2 in the case of multiclass categorization

- Visualizing the Impact of K: Figures 4 and 5 give us a visual look at how the model performed with the CIC-IDS2017 data. It consistently shows high detection rates across different attack types, a good sign of robust implementation. Although raising the K-Fold value led to minor variations in detection rates, the SFH-Partial approach consistently maintained minimum false alarm rates throughout

Figure 6 shows the detection rate for different kinds of attacks in the UNSW.NB15. This model generally performed well, especially for attack types with plenty of examples. However, Worm and DoS attacks proved trickier, with

detection rates dropping close to zero as K increased. Based on the confusion matrix, the model often misclassified these as reconnaissance attacks. Figure 7 zooms in on the FAR values for these same attack types, showing that DoS attacks had the highest FAR. Finally, Figures 8 and 9 illustrate the results from the WSN-DS data. Found out that increasing the no. of K-Folds improved the detection of Black hole attacks but hurt the detection of Grayhole attacks. Other attack types showed similar detection rates across different K-Fold values. Overall, the results weren't as good as expected, so we improved the model at catching all kinds of attacks.

4.5 CNN-LSTM Performance: Epoch Analysis

After assessing the influence of K-Fold cross-validation, attention was turned to evaluating the impact of varying the number of training epochs. Building upon prior experiments, we maintained a K-value of 8 for these tests. After assessing the influence of K-Fold cross-validation, attention was turned to evaluating the impact of varying the number of training epochs. Building upon prior experiments, we maintained a K-value of 8 for these tests. Figures 10 & 11 detail the observed effects of increasing the epoch count on each detection rate and False Alarm Rate for binary classification tasks. Findings indicate that the data of UNSW-NB15 was susceptible to several epochs. As depicted in Figure 10, the detection rate improved from 94.53% at five epochs to 95.81% at sixty epochs. In contrast, the CIC-IDS2017 dataset exhibited accuracy values of 99.7% and 99.93% at 5 and 60 epochs, respectively, while WSN-DS showed 98.14% and 97.86%. Examination of FAR values (Figure 11) reveals that UNSW-NB15, compared to the other datasets, yielded the highest FAR values across the tested epoch range
Figures 12 and 13 illustrate the performance of multiclass & binary classification. The UNSW.The NB15 dataset consistently demonstrated the poorest detection performance and maximum false alarm rates. Additionally, the figures indicate that increasing the no. of epochs had minimal impact on the performance of each CIC-IDS and WSN-DS dataset.
The confusion matrices for the three datasets reveal that the model generally achieved accurate classification across most record types. However, a notable trend was observed in CIC-IDS: PortScan attacks were frequently misclassified as regular traffic. In UNSW-NB15: Exploits, Fuzzers, DoS, and Worms attacks were often misclassified as Reconnaissance attacks. Conversely, WNS-DS demonstrates the capability of the model to accurately classify all kinds of records in the dataset, accurately predicting most of the entries in each group.

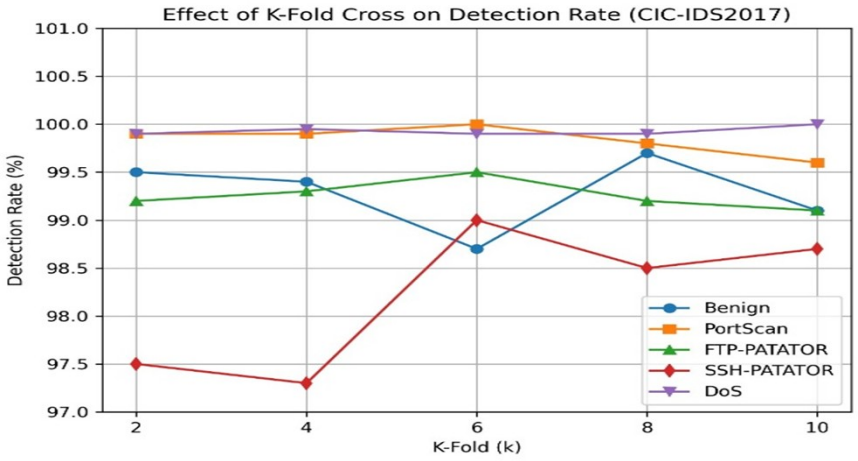

Figure 4: K-Fold impact on detection accuracy (CIC-IDS)

5 System Performance Review

To ascertain the efficacy of this hybrid model, its accuracy was benchmarked against existing cutting-edge methodologies. The results, summarized in the subsequent tables, illustrate that the model attains a superior level of overall performance when compared to other recent investigations. Specifically, a comparative analysis was performed utilizing a dataset configured with 5 epochs, a K value of 8, and a focus on binary classification. Starting with the

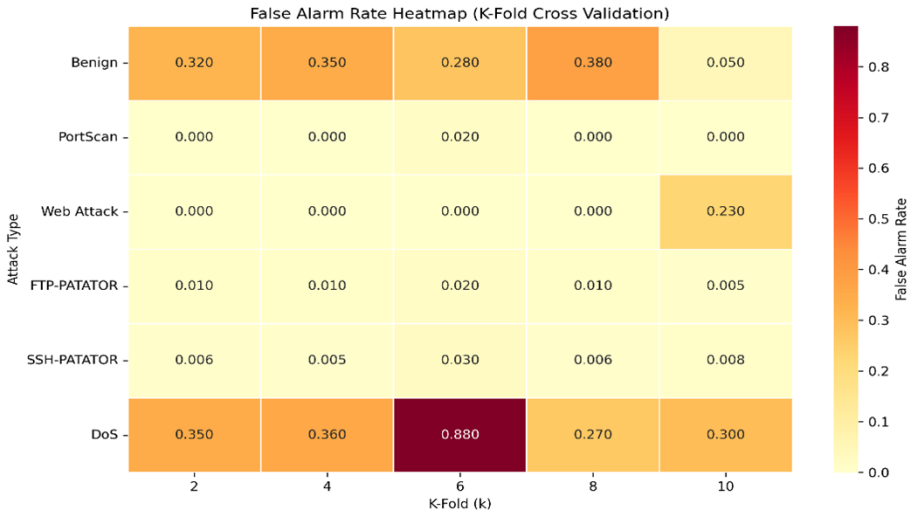

Figure 5: - K-Fold impact on FAR (CIC-IDS)

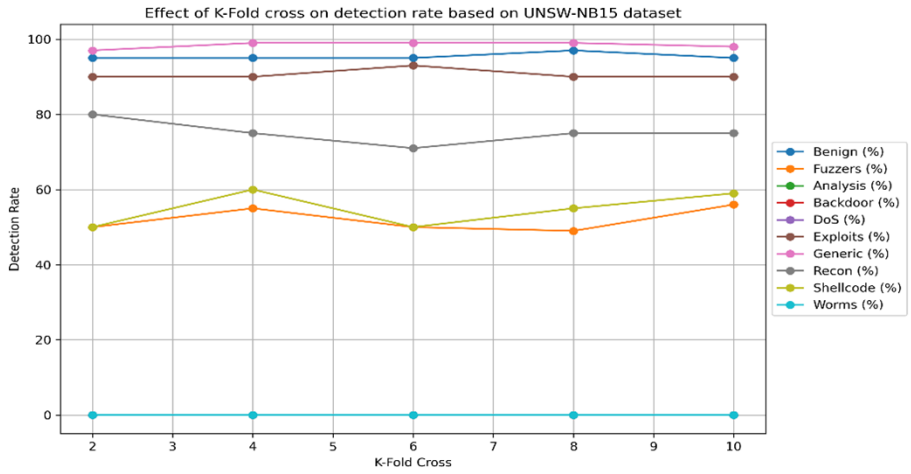

Figure 6: K-Fold impact on detection accuracy (UNSW.NB15)

dataset UNSW.NB15, the model showed improved performance compared to various machine learning and deep learning models. The CNN+LSTM-based model achieved a 93.78% accuracy rate, which is markedly higher than the 85.77% of the Deep Belief Network (DBN), and then with the Deep Neural Network (ICVAE-DNN) achieving 82.42% accuracy, as well as higher than the Support Vector Machine model [4]. Notably, CNN+LSTM also exhibited the minimum False Alarm Rate (FAR), yielding similar results. While the detection rate was marginally lower than that of other specific models, the collective performance substantiates the superiority of the CNN & LSTM layer stacking approach when evaluated against the UNSW-NB15 dataset.

Regularization through dropout and normalization techniques contributed towards improved model Broader applicability and stability, as the classifier trained on CIC-IDS for binary tasks demonstrates considerable robustness. The convolutional and recurrent hybrid model architecture achieved an impressive precision of 99.65%, exceeding the Fully Connected Neural Network (FCNN) results. Compared to Rep Tree (96.68%) and kNN (80.17%), the model achieved 85.24%. Moreover, the CNN+LSTM model showcased better false alert frequency and detection performance than other assessed models. Performance outcomes using the WSN-DS dataset. The proposed model achieved a correctness rate of 99.59%, outperforming alternative Learning algorithms, for example, Logistic Regression (LR) at 97%, Naïve Bayes at 83.20%, and Decision Tree (DT) at 99.20%. The CNN+LSTM combined model also provides a peak detection rate of 97.78%. These results highlight the efficiency of this model, which can

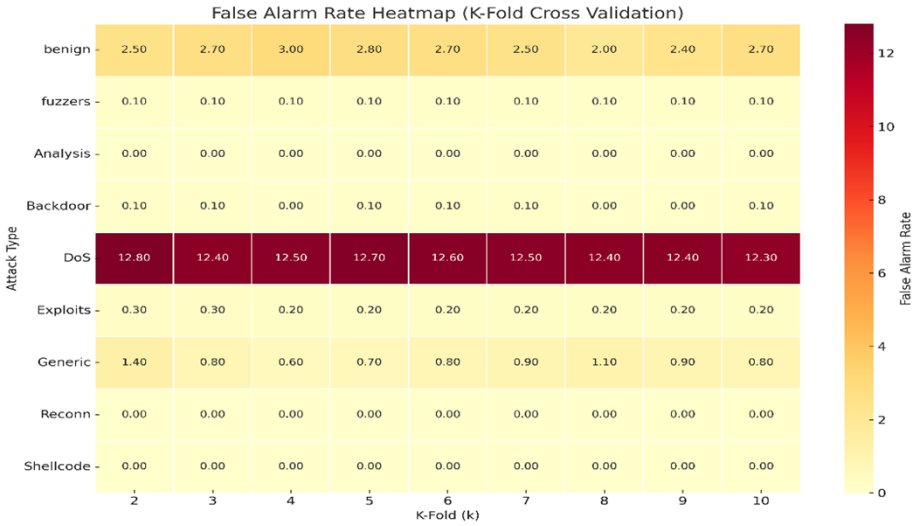

Figure 7: K-Fold impact on FAR (UNSW.NB15)

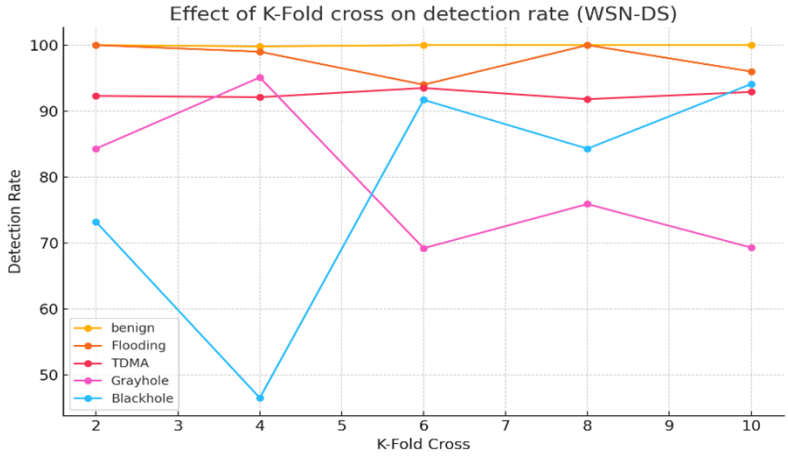

Figure 8: K-Fold impact on detection accuracy (WSN-DS)

be attributed to the cascaded CNN & LSTM layers followed by DNN, meticulous dataset preprocessing, optimized feature selection, Incorporating dropout regularization, and batch normalization methods.

6 Discussion

This work proposed building a much more powerful intrusion detection system (IDS) for detecting benign and malicious traffic with greater accuracy. An essential challenge in this space is that there have been many attacks in recent years, and IDS systems must evolve to stay relevant. Existing IDSs yield many false positives that can be overwhelming and lead to incomplete detection of actual threats. Furthermore, training data accumulated in previous research may become outdated and thus less predictive of new attack vectors. To address these challenges, a new system architecture was introduced that merges CNN's feature extraction abilities and the temporal dependency modelling of Long Short-LSTM networks. This combined approach uses two different ways of looking at network traffic data. The three- The layer model uses CNNs and LSTMs to find intrusions while keeping false alarms to a minimum. Before training, the data went through a careful preprocessing stage, including encoding, normalization, and feature selection, to help the model learn better. This refined data is then fed into the first CNN layer, which is good at picking out temporal patterns. The system is tested using the CIC-IDS2017 dataset over five training

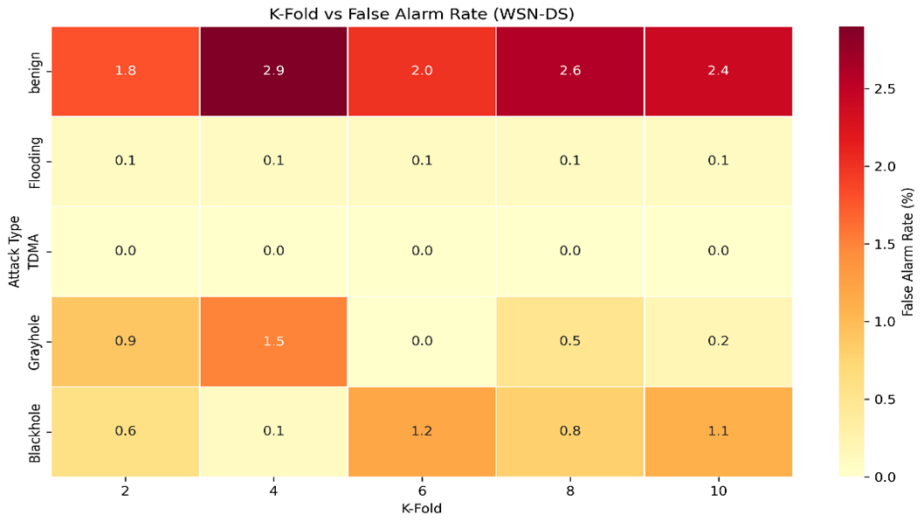

Figure 9: K-Fold impact on FAR (WSN-DS

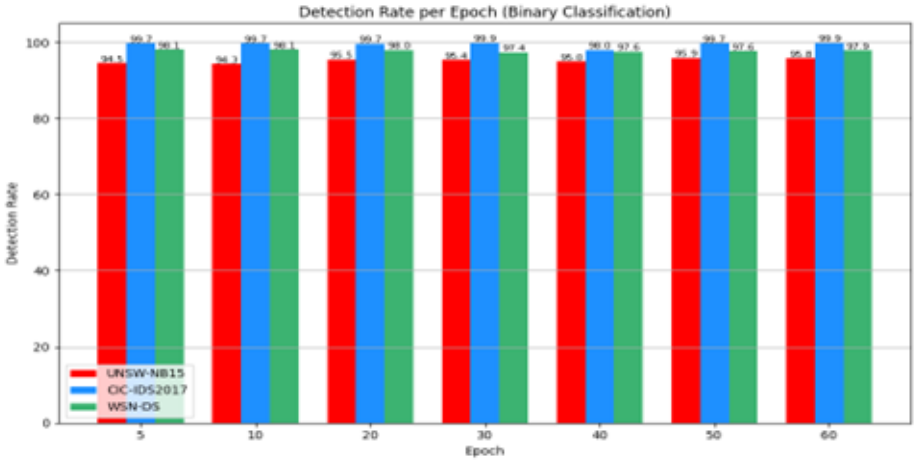

Figure 10: Effect of epoch count on binary classification detection

cycles. The results were auspicious. When classifying traffic as either an attack or not, the system achieved an accuracy of about 99.66%. When identifying specific types of attacks, the accuracy was still high at 99.61%. The accuracy and F1-scores confirmed the model's accuracy & reliability, with an attack detection rate of 99.71% in binary classification and 99.96% in multiclass classification. The false positive rates were relatively low, at 0.10% and 0.12% in binary and multilabel classification scenarios. The system's ability to work using the dataset UNSW-NB15 was also evaluated. The results were pretty good! It gives about 94.54% accuracy when classifying things into two categories (binary classification) and around 82.42% when classifying into multiple categories (multiclass classification). The WSN-DS was also utilised as another way to estimate performance. Interestingly, the results with this dataset were similar to what was evaluated with the CIC-IDS2017 dataset. It was highly accurate, detected things well, and didn't give us many false alarms. In conclusion, these experimental results suggest that this hybrid CNN+LSTM model offers a practical and effective way to develop more accurate and reliable IDS.

7 Conclusion and future works

In this study, a tool was introduced that uses spatial and sequential deep learning algorithms (LSTM and CNN) to identify suspicious activity in networks. The model uses CNN's capacity to extract spatial attributes and LSTM's ability to describe time-based patterns by integrating the CNN and LSTM model layers. Different techniques were

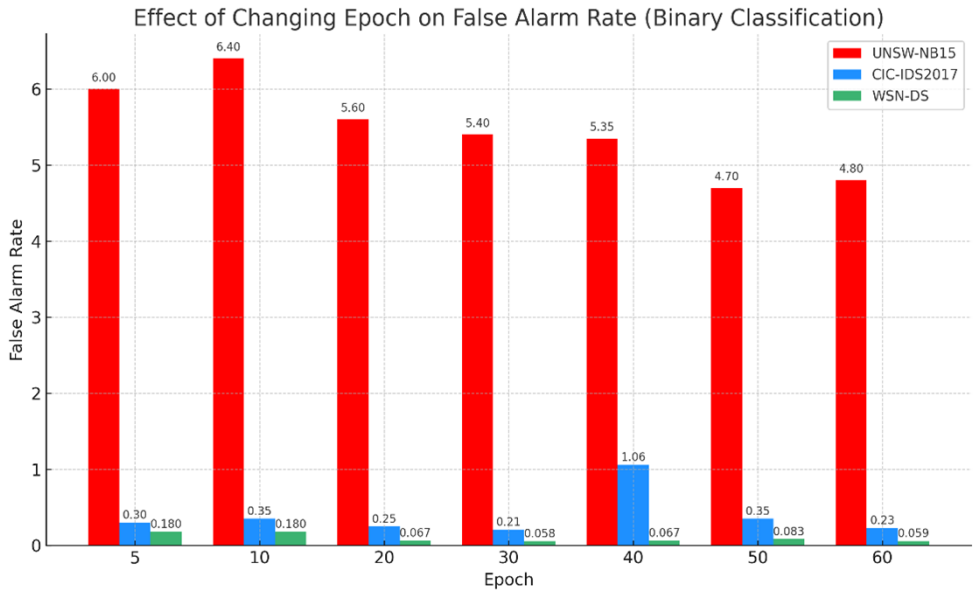

Figure 11: Effect of epoch count on binary classification FAR

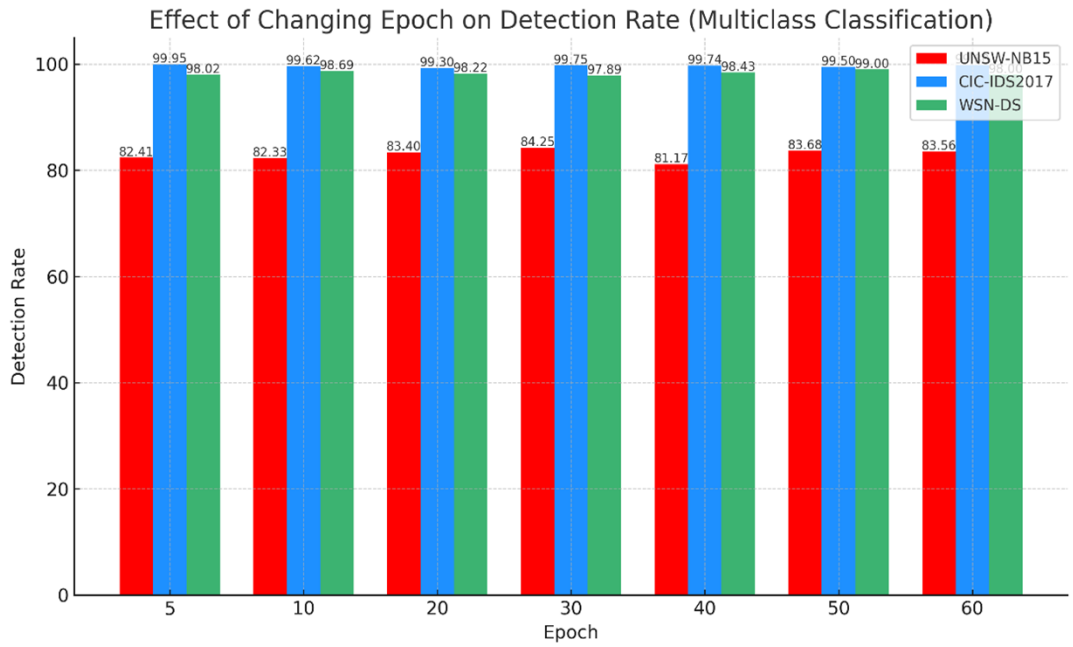

Figure 12: Effect of epoch count on multiclass classification detection

used, like feature standardization, regularization through dropout, and batch wise normalization to increase the efficacy of the architecture. Three benchmark datasets—the UNSW-NB15, the CIC-IDS, and the Wireless Sensor Networks (WSN) Data Set, each including malicious and non-harmful traffic data, were used to train and evaluate the system. Individual and hybrid models such as Convolutional Neural Networks, Long Short-Term Memory networks, and CNN+LSTM architectures were used to assess how these datasets behaved. After that, two-class and multiple class classification tasks were used to evaluate the merged model. For the CIC-IDS, Wireless Sensor Network (WSN) and UNSW.NB15 datasets, the model achieved binary classification accuracy scores of 99.65%, 94.54%, and 99.68% during 5 training epochs. Overall identification and false alarm rates were encouraging, despite less successful performance against specific threat types, such as web-based attacks in CIC-IDS and worms or backdoor attacks in UNSW-NB15. Furthermore, the impact of increasing training epochs and cross-validation of

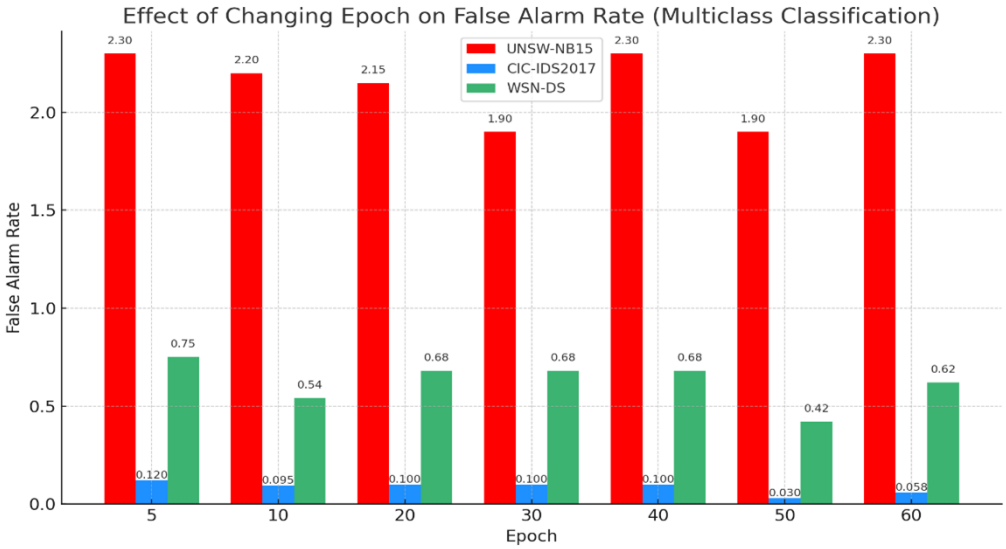

Figure 13: Effect of epoch count on multiclass classification FAR

Table 4: Performance comparisons across the datasets

Dataset	**Method**	**Accuracy (%)**	**FAR (%)**	**Detection Rate (%)**
UNSW-NB15	SVM	62.43	*	88.59
UNSW-NB15	ICVAE-DNN	89.09	19.02	95.69
UNSW-NB15	DBN	85.78	30.33	98.91
UNSW-NB15	CNN-LSTM	93.79	6.01	94.54
CIC-IDS	KNN	80.92	*	91.29
CIC-IDS	REP Tree	96.68	1.15	94.48
CIC-IDS	MLP	85.25	7.36	77.84
CIC-IDS	CNN-LSTM	**99.65**	**0.11**	**99.71**
WSN-DS	LR	97.00	*	77.71
WSN-DS	NB	83.11	*	76.51
WSN-DS	DT	99.11	*	95.11
WSN-DS	CNN-LSTM	**99.59**	*	**97.78**

K-Folds was examined. The model's performance first improved before plateauing, according to the results. By tackling issues related to poor detection rates and high FARs brought on by class imbalance in the datasets, subsequent developments will prioritize refining detection capabilities and minimizing false alarms.

References

[1] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[2] N. Faruqui, M. A. Yousuf, M. Whaiduzzaman, A. Azad, S. A. Alyami, P. Liò, M. A. Kabir, and M. A. Moni, "Safetymed: A novel iomt intrusion detection system using cnn-lstm hybridization," *Electronics*, vol. 12, no. 17, p. 3541, 2023.

[3] M. Abdallah, N. An Le Khac, H. Jahromi, and A. Delia Jurcut, "A hybrid cnn-lstm based approach for anomaly detection systems in sdns," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–7.

[4] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Cnn-lstm: hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99 837–99 849, 2022.

[5] H. Alkahtani and T. H. Aldhyani, "Botnet attack detection by using cnn-lstm model for internet of things applications," *Security and Communication Networks*, vol. 2021, no. 1, p. 3806459, 2021.

[6] H. Sun, M. Chen, J. Weng, Z. Liu, and G. Geng, "Anomaly detection for in-vehicle network using cnn-lstm with attention mechanism," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 880–10 893, 2021.

[7] R. Jablaoui and N. Liouane, "An effective deep cnn-lstm based intrusion detection system for network security," in *2024 International Conference on Control, Automation and Diagnosis (ICCAD)*. IEEE, 2024, pp. 1–6.

[8] A. Taneja and G. Kumar, "Attention-cnn-lstm based intrusion detection system (acl-ids) for in-vehicle networks," *Soft Computing*, vol. 28, no. 23, pp. 13 429–13 441, 2024.

[9] M. K. Putchala, "Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru)," Master's thesis, Wright State University, 2017.

[10] B. Deore and S. Bhosale, "Hybrid optimization enabled robust cnn-lstm technique for network intrusion detection," *Ieee Access*, vol. 10, pp. 65 611–65 622, 2022.

[11] A. Gueriani, H. Kheddar, and A. C. Mazari, "Enhancing iot security with cnn and lstm-based intrusion detection systems," in *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, 2024, pp. 1–7.

[12] S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, "A hybrid cnn-lstm approach for intelligent cyber intrusion detection system," *Computers & Security*, vol. 148, p. 104146, 2025.

[13] P. Rajak, J. Lachure, and R. Doriya, "Cnn-lstm-based ids on precision farming for iiot data," in *2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*. IEEE, 2022, pp. 99–103.

[14] V. Poornachander, K. S. Kumar, and S. Jagadish, "Ddos attack intrusion detection system with cnn and lstm hybridization," in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. IEEE, 2024, pp. 1–6.

[15] N. S. Bhati, M. Khari, V. García-Díaz, and E. Verdú, "A review on intrusion detection systems and techniques," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 28, no. Supp02, pp. 65–91, 2020.

[16] H. Sharma, P. Kumar, and K. Sharma, "Recurrent neural network based incremental model for intrusion detection system in iot," *Scalable Computing: Practice and Experience*, vol. 25, no. 5, pp. 3778–3795, 2024.