IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 August 2022, accepted 2 September 2022, date of publication 14 September 2022, date of current version 26 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206425

RESEARCH ARTICLE

CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System

ASMAA HALBOUNI[1], (Member, IEEE), TEDDY SURYA GUNAWAN[1], (Senior Member, IEEE), MOHAMED HADI HABAEBI[1], (Senior Member, IEEE), MURAD HALBOUNI[2], MIRA KARTIWI[3], (Member, IEEE), AND ROBIAH AHMAD[4], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia
[2]Department of Natural, Engineering and Technology Sciences, Arab American University, Jenin 240, Palestine
[3]Department of Information Systems, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia
[4]Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia

Corresponding author: Teddy Surya Gunawan (tsgunawan@iium.edu.my)

This work was supported by the Ministry of Higher Education (MOHE), Malaysia, under the Fundamental Research Grant Scheme (FRGS) Vote FRGS/1/2019/TK10/UTM/02/16.

ABSTRACT Network security becomes indispensable to our daily interactions and networks. As attackers continue to develop new types of attacks and the size of networks continues to grow, the need for an effective intrusion detection system has become critical. Numerous studies implemented machine learning algorithms to develop an effective IDS; however, with the advent of deep learning algorithms and artificial neural networks that can generate features automatically without human intervention, researchers began to rely on deep learning. In our research, we took advantage of the Convolutional Neural Network's ability to extract spatial features and the Long Short-Term Memory Network's ability to extract temporal features to create a hybrid intrusion detection system model. We added batch normalization and dropout layers to the model to increase its performance. Based on the binary and multiclass classification, the model was trained using three datasets: CIC-IDS 2017, UNSW-NB15, and WSN-DS. The confusion matrix determines the system's effectiveness, which includes evaluation criteria such as accuracy, precision, detection rate, F1-score, and false alarm rate (FAR). The effectiveness of the proposed model was demonstrated by experimental results showing a high detection rate, high accuracy, and a relatively low FAR.

INDEX TERMS Intrusion detection system, deep learning, convolutional neural network, long-short term memory, accuracy, false alarm rate, binary classification, multiclass classification.

I. INTRODUCTION

The rapid growth of technologies and information, such as the internet of things, big data, and cloud computing, as well as the increasing reliance of our daily communications on networked services, have made networked computing essential, thereby increasing the significance of network security. Any vulnerability or threat will affect the entire network [1]. Firewalls and encryption techniques are traditional security mechanisms that face challenges where the attackers keep developing complicated attacks [2]. Moreover, cybersecurity researchers found the importance of developing efficient network intrusion detection systems (IDS) to provide secured networks. Intrusion detection systems intend to provide availability, confidentiality, and integrity for the data transmitted in networked computers by preventing unauthorized access to a network, protecting the information and communication systems in the network [3], and, most important, being able to detect known and unknown attacks and threats with high accuracy and a minimum false alarm rate [4].

Two approaches comprise the intrusion detection system: misuse detection and anomaly detection. Misuse detection, also known as signature-based detection, is the initial detection model where detection is based on known and stored attacks and threats. This model has a low rate of false alarms and a high detection rate. With the expansion of networks

The associate editor coordinating the review of this manuscript and approving it for publication was Nazar Zaki.

and services, unknown new attacks are being developed by attackers, which makes the model susceptible to these attacks [5]. To provide security for these networks, an intrusion detection system must be effective and intelligent in detecting and preventing known and unknown attacks, such as anomaly detection. Despite a high false alarm rate, anomaly detection can detect known and unknown attacks.

Artificial Intelligence (AI) has made it possible for computers and machines to learn from a dataset with minimal human intervention; intrusion detection systems have taken advantage of this capability. Both machine learning (ML) and deep learning (DL) are sub-fields of artificial intelligence (AI), and both were utilized in the creation and development of an effective intrusion detection system. The classification and detection of network traffic in a machine learning system are based on manually extracted features. While the deep learning system, with its neural network, can extract features from the dataset and then perform classification and detection, deep learning can enhance and improve the detection accuracy of the model in comparison to machine learning [1].

Based on various approaches and learning techniques, numerous models have been and continue to be developed to create an effective intrusion detection system. Existing models have poor precision, low detection, and high false alarm rates. In this paper, we attempt to address these concerns and develop a more effective detection model. This paper proposes a deep learning-based intrusion detection system that employs two deep learning algorithms, Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM). Both algorithms extract temporal and spatial features of network traffic to reduce the false alarm rate and increase the detection rate. This paper evaluated the model using the CIC-IDS2017, UNSW-NB15, and WSN-DS datasets and compared the outcome to the CNN model, LSTM model, and other learning algorithms. Our results demonstrated that integrating two deep learning algorithms will improve the detection rate and accuracy, making the model more accurate and resistant to threats and attacks.

This paper is structured as follows: Section 2 provides an overview of machine learning for network intrusion detection. Our hybrid CNN-LSTM model is structured in Section 3. Section 4 describes the dataset utilized in the model's development. Section 5 describes the experimental design and evaluation of the model, while Section 6 concludes this paper.

II. INTRUSION DETECTION SYSTEM

Cybersecurity researchers are attempting to create a model that can detect known and unknown network attacks and prevent them from causing damage to the network. As will be demonstrated next, the algorithms developed for IDS can be divided into machine learning and deep learning.

A. MACHINE LEARNING-BASED IDS

Machine learning played and still plays a vital role in intrusion detection systems. ML algorithms are based on supervised learning, such as Decision Tree, SVM, and Naïve Bayes, and unsupervised learning, such as K-means clustering and Self Organized Map [4]. The primary function of machine learning algorithms is to enhance a system's detection capability. The trained data is used to detect attacks and threats. Machine learning algorithms are typically employed to solve regression, classification, and clustering problems. Most prior work on machine learning relied on the NSL-KDD, DARPA, and KDD-CUP99 datasets. Some models produced satisfactory results, but these datasets are out-of-date and contain only simple types of attacks [1], [4]. Training an IDS for the current, continuously expanding network requires a large dataset, and relying on traditional machine learning algorithms that function correctly on small datasets will not result in an efficient model [4].

B. DEEP LEARNING-BASED IDS

Deep learning is a subfield of machine learning that interacts with multi-hidden-layer artificial neural networks [4]. In addition to data representations, deep learning algorithms can also learn from unlabeled or unstructured data [6]. Deep learning has many performance features that allow it to be efficient enough to develop an IDS, such as the robustness of the DL algorithms with high scalability and the ability to deal with different types of data [7]. Deep learning algorithms were mainly developed to solve complex problems, pattern recognition, search engine, and machine translation [8]. Algorithms such as Deep belief networks (DBN), Restricted Boltzmann machines (RBM), and Autoencoder (AE) are used widely for extracting features [9]. Multi-Layer Perceptron is used in different fields and mainly to minimize the error rate during training [10].

Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are the most prevalent deep learning algorithms. CNN's primary advantage is its ability to automatically recognize spatial features without human intervention, avoid overfitting by reducing the number of trainable parameters, and improve generalization [8]. RNN is primarily used in Natural Language Processing (NLP), speech processing, and video analysis due to its ability to utilize sequential network features [7], [11]. Due to the memory blocks in RNN's neural network, LSTM was developed as a solution to the RNN's vanishing gradient problem [11].

III. METHODOLOGY

In our research, we construct an intrusion detection system using CNN-LSTM layers. The IDS model's methodology is depicted in Figure 1.

A. DATASET PREPARATION

The initial step in constructing an effective intrusion detection system is to select an appropriate dataset. The dataset should include normal and malicious records representing what the model will encounter in the real world. Our research uses the CIC-IDS2017, UNSW-NB15, and WSN-DS datasets, all of which are newly accessible. These datasets contain normal

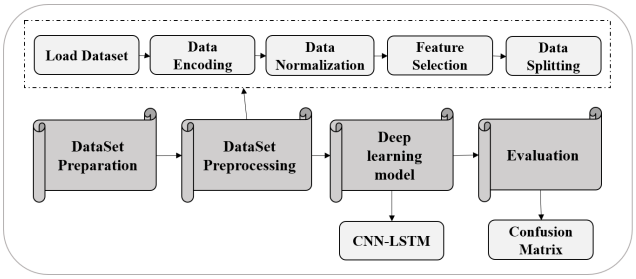

FIGURE 1. Methodology flowchart.

and malicious traffic data that is regarded as new and does not contain a significant amount of redundant information.

1) CIC-IDS2017

CIC-IDS2017 encompasses eleven new attacks, including Brute Force, PortScan, DoS, web attacks including XSS and SQL Injection, FTP-Patator, and SSH-Patator. It was developed in 2017 by the Canadian Institute for Cybersecurity, and its eighty features are used to monitor benign and malicious traffic [6], [12].

2) UNSW-NB15

This dataset contains records of benign traffic and nine types of attacks, such as Fuzzers, Analysis, Backdoor, DoS, Exploits, etc. The Australian Centre for Cyber Security (ACCS) created it in 2015. The records were collected from three real-world websites, including BID (Symantec Corporation), CVE (Common Vulnerabilities and Exposures), and MSD (Microsoft Corporation) (Microsoft Security Bulletin) [4], [12].

3) WSN-DS

WSN-DS was developed in 2016 to detect normal and malicious traffic by monitoring the number of nodes in wireless networks with sensors. This dataset's records are extracted using the LEACH routing protocol, represented by 23 features. There are standard records and four DoS attack types, including flooding, Grayhole, blackhole, and TDMA [13].

B. DATA PREPROCESSING

1) LOAD DATASETS

The datasets we used were publicly available. The data is stored in a CSV file in pcap format. In this step, Pandas package was used to read each dataset's details, and after reading each dataset's details, it was cleaned of any null and duplicate values in preparation for the next step.

2) DATA ENCODING

This step is responsible for encoding labels in datasets. Dealing with a deep neural network means dealing with numerical values. Labels in each dataset are not numerical values, so by using the One-Hot Encoder, we encoded the label column by changing the values from benign or malicious to be represented by numerical values.

3) DATA NORMALIZATION

Normalizing the data is a preprocessing technique used to optimize within-range characteristics. The variance of the data read from the CSV file, which has different standard derivations and means, will impact the learning efficiency. In our model, we scaled the input data using Standard Scalar, resulting in a mean of zero and a standard deviation of one. Based on 'sklearn. preprocessing' library Standard Scalar was used to normalize the datasets.

4) FEATURE SELECTION

Feature selection is also referred to as feature reduction and is responsible for selecting a set of features based on criteria. This process enables rapid model construction and training based on specific features, which reduces training and testing time and improves performance. In our work, we used a method called SelectKBest. SelectKBest was imported from the 'sklearn. feature selection' library which selects the best features based on the highest score. We chose the source function to perform classification and the number of features based on K values. The output is an array containing the score and the name of the feature, and we chose our features based on that array

5) DATA SPLITTING

Our model's datasets have been divided into 80% training and 20% testing set. In addition, we divide the training set into training and validation sets to tune our hyperparameters during training to improve the model's performance. Using the Stratified K-Fold Cross Validation technique, the size of both sets was determined based on the factor K.

C. HYBRID DEEP LEARNING MODEL

CNN can extract spatial features, while LSTM can extract temporal characteristics. Due to CNN's ability to extract high-level features from large amounts of data, the model begins with CNN. The first layer is the CNN layer; the data will then pass through the convolution layer, where the filters will extract the most critical features to generate a feature map. This map will undergo max pooling to preserve the most dominant features, followed by batch normalization. The output will be sent to an LSTM layer to extract temporal features, followed by a dropout layer to prevent overfitting. This combination of CNN and LSTM layers will be repeated three times with varying numbers of neurons and filters, followed by a fully connected layer that uses the SoftMax activation function to perform classification. Figure 2 depicts the structure of our deep learning model.

1) CONVOLUTIONAL NEURAL NETWORK

CNN has two components: convolution and pooling. The convolution layer applies a set of filters through a mathematical

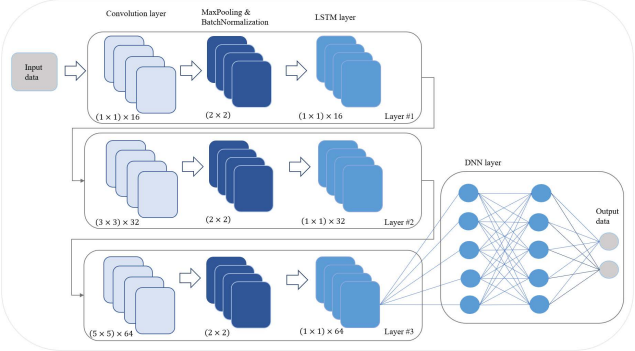

FIGURE 2. CNN-LSTM layers structure.

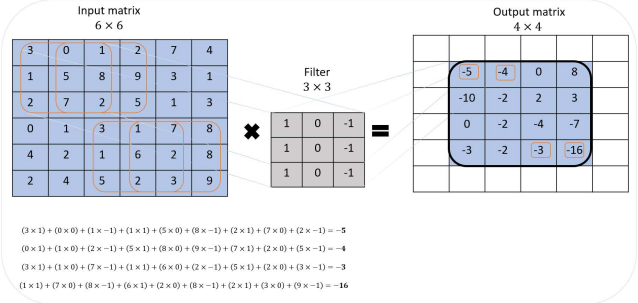

FIGURE 3. Convolution process.

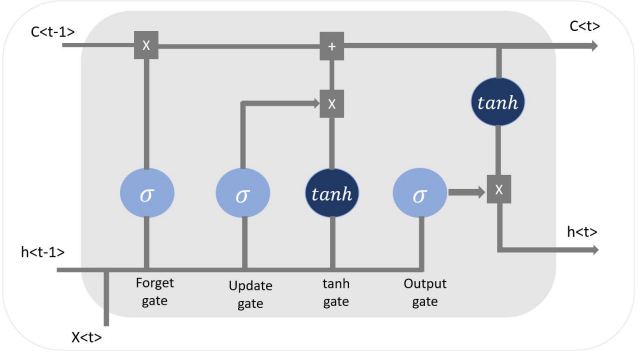

FIGURE 4. LSTM structure.

operation. The process involves applying the filter to the input matrix to produce the feature map. It begins when the kernel slides over the input matrix in horizontal and vertical directions. At this point, the dot product between the input matrix and the kernel is calculated based on the multiplication of their elements and then summed into a single scalar value; this process is repeated until sliding is no longer possible. These new output matrix values represent the feature map. A threshold-based activation function will process the feature map to determine whether the neuron will fire or not [6], [7]. In our model, we used ReLU as an activation function, as follows: $ReLU(z_i) = \max(0, z_i)$. Therefore, the equation after the activation function will be:

$$Z = h\left(\sum_{i}^{p\times q} w_i v_i + b\right) \tag{1}$$

where h represents the activation function, w is the weights, v is the input data, b is the bias, and p and q are the sizes of the input data matrix. Then pooling layer comes after performing the convolution on the data. The purpose of the pooling layer is to decrease the generated matrix's size to prevent overfitting and enhance learning. The Max pooling technique will reduce the sample size without affecting the weights [14].

2) BATCH NORMALIZATION

Batch Normalization (BN) is primarily used to avoid covariance shifts resulting from changing the input from one layer to the next layer in a deep neural network, as these shifts make the learning process unstable and reduce the learning efficiency. BN will accelerate the optimization procedure and reduce generalization errors [15]. In addition, it will adjust CNN output by scaling the data in the input layer to a unit norm, followed by LSTM layer processing. The mathematical representations of batch normalization are in the formulas next. X represents the data generated from the Max pooling layer, μ_B and δ_B are mean and variance of batch, respectively, $\in$ to ensure that the denominator in the formula is non-zero.

$$\hat{X} = \frac{X - \mu_B}{\sqrt{\delta_B^2 + \epsilon}} \tag{2}$$

The result of equation 2 will be processed with two variables γ and β. This process will generate an output $\hat{Y}$, where γ and β are used for better learning output by training them in the learning process.

$$\hat{Y} = \gamma\hat{X} + \beta \tag{3}$$

3) LONG-SHORT TERM MEMORY

The central concept of LSTM is its capacity to translate and cache inputs using memory cells over time. This memory cell will be processed by gates whose activation function is represented by gates. As shown in Figure 4, LSTM consists of four gates: forget gate, update gate, tanh gate, and output gate. In these networks, the learning process occurs by adjusting the weights and the value of the activation function so that the temporal features between input and output data can be effectively produced [3], [16], [17].

In the LSTM network, input and output values are the vectors of the same size set by $X(t)$. Forget gate will decide which information to keep and which to delete by combining $X(t)$ with the previously hidden state $X(t-1)$. Moreover, the output will be generated based on the sigmoid function and multiplied with the previous cell state $C(t-1)$. The update gate considers the input gate, which will determine

TABLE 1. Confusion matrix.

	Predicted as Positive	Predicted as Negative
Labeled as Positive	True Positive (TP)	False Negative (FN)
Labeled as Negative	False Positive (FP)	True Negative (TN)

the information needed to be added to generate $C(t)$. This generation will be based on the sigmoid function and tanh function based on tanh gate. The multiplication of these gates will be added to the output resulting from multiplying forget gate with $C(t-1)$ to generate $C(t)$. The current cell state $C\ (t)$ goes through tanh activation function and then multiplied by the output of the sigmoid activation function of the output gate to generate the currently hidden state $h(t)$ representing the output of the LSTM network. The following equation represents the formula of the output:

$$O\,(t) = \sigma\,(b + U \times X\,(t) + W \times h(t-1)) \tag{4}$$

4) DROPOUT

Neurons are dropped randomly during the training process in each epoch using this technique [9]. This process is necessary for deep neural networks to prevent overfitting, in which the network learns too well, limiting its capacity to identify variables in new samples [8]. In our research, we added a layer with a 0.2 dropout rate.

5) FULLY CONNECTED LAYER

The final layer operates on the extracted map features. Fully connected (FC) means that each neuron in this layer is connected to all neurons in the layer beneath it. This layer is responsible for implementing classification, performed using the Softmax activation function [3], [9]. The input data will be transformed into a one-dimensional layer to classify the data into the appropriate class and assign output probabilities, with the output from this layer representing the final output.

D. EVALUATION

The confusion matrix indicators, as shown in Table 1, are used to evaluate the performance of IDS. *TP* represents benign records incorrectly classified as malicious, *FP* represents benign records incorrectly classified as malicious, *TN* represents malicious records incorrectly classified as benign, and *FN* represents malicious records incorrectly classified as benign.

From the confusion matrix indicators, we obtain accuracy (*ACC*), detection rate (*DR*), precision (*Pr*), and false alarm rate (*FAR*). *ACC* refers to the ratio of true predictions of the records. *DR* is the ability to predict only positive records in their entirety. *Pr* is the ability to avoid mislabeling negative records as positive, whereas *FAR* is the ratio of normal traffic misclassifications.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

TABLE 2. Experimental scenario.

DATASET	CLASSIFICATION			
	Binary		Multiclass	
	No. of records	Type of records	No. of records	Type of records
CIC-IDS 2017	2	Normal and malicious	6	Normal, web attack, SSH-Patator, FTP-Patator, and PortScan
UNSW-NB15	2	Normal and malicious	10	Normal, Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, shellcode, worms
WSN-DS	2	Normal and malicious	5	Normal, Flooding, TDMA (scheduling), Grayhole, Blackhole

TABLE 3. Accuracy of CIC-IDS2017 binary classification based on different learning algorithms.

CIC-IDS 2017 Binary classification (%)				
No. of layers	CNN	LSTM	CNN-LSTM	LSTM-CNN
1	97.42	98.89	98.40	99.02
2	99.15	99.19	99.49	99.20
3	98.95	99.01	**99.59**	99.56

$$DR = \frac{TP}{TP + FN} \tag{6}$$

$$FAR = \frac{FP}{FP + TN} \tag{7}$$

$$Pr = \frac{TP}{TP + FP} \tag{8}$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

We constructed our model on an evaluation platform comprised of a Dell Inspiron 15 3511 with an Intel(R) Core (TM) i7-1165G7 processor running at 2.80 GHz and 8.00 GB of RAM. The model for deep learning was implemented using the TensorFlow, Pandas, and Keras libraries.

We have evaluated the models using two classification methods: binary and multiclass. The datasets were divided into two classes for binary classification: benign and attack. As shown in Table 2, the dataset is labeled as benign or as one type of attack for multiclass classification.

1) COMPARISON BASED ON DIFFERENT LEARNING ALGORITHMS

In the initial phase of our research, we compared the performance of datasets based on CNN-alone, LSTM-alone, LSTM-CNN, and CNN-LSTM to determine which model provided the best results. The outcomes are presented in Tables 3, 4, and 5.

Table 3 shows the accuracy of the CIC-IDS 2017 binary dataset. The highest accuracy achieved by CNN-LSTM structures with three layers was 99.59 %, followed by LSTM-CNN

TABLE 4. Accuracy of UNSW-NB15 binary classification based on different learning algorithms.

UNSW-NB15 Binary classification (%)				
No. of layers	CNN	LSTM	CNN-LSTM	LSTM-CNN
1	93.10	93.18	93.62	93.49
2	93.19	93.36	93.65	93.58
3	93.65	93.34	**93.68**	93.67

TABLE 5. Accuracy of WSN-DS binary classification based on different learning algorithms.

WSN-DS Binary classification (%)				
No. of layers	CNN	LSTM	CNN-LSTM	LSTM-CNN
1	99.53	99.62	99.58	**99.64**
2	99.34	99.34	99.53	99.60
3	99.60	99.60	99.61	99.59

TABLE 6. Feature Selection based on binary CIC-IDS2017.

CIC-IDS 2017 Binary classification (%)						
No. of feature	ACC	Pr	DR	F1-score	FAR	Training Time (s)
24	97.32	95.87	99.0	95.80	0.83	763
40	99.40	99.60	99.30	99.60	0.45	322
50	99.59	99.58	**99.54**	99.62	**0.12**	467
60	**99.60**	99.30	99.0	99.3	0.3	472
78	99.56	99.60	99.52	99.52	0.13	2070

structures with three layers at 99.56 %. Finally, CNN-LSTM structures with two layers at 99.49 %.

The results for UNSW-NB binary dataset are in Table 4. Three layers of CNN-LSTM achieved the highest accuracy at 93.68 %, followed by three layers of LSTM-CNN at 93.67%, three layers of CNN-only at 93.65 %, and two layers of CNN-LSTM at 93.65 %.

WSN-DS shows distinct behavior. The LSTM-CNN structure with a single layer structure achieved the highest accuracy of 99.64%, followed by CNN-LSTM with 99.61% for three layers and 99.62% for one layer of LSTM, as shown in Table 5. After comparing four learning algorithms, we continued our research using the CNN-LSTM hybrid structure.

2) CNN-LSTM BASED ON SELECTED FEATURES

The second phase of our testing involved the selection of model-building features. Initially, utilizing the CIC-IDS2017 dataset and only one layer of CNN-LSTM, we conducted five experiments with 24, 40, 50, 60, and 78 features. We conducted three experiments with 24, 32, and 42 features for UNSW-NB15. Based on WSN-DS, we tested six, twelve, and eighteen features; the results are presented in the following tables. The selection of features was determined by SelectKBest, which selected the highest score.

The results based on the binary CIC-IDS2017 dataset are displayed in Table 6. 24 features scored 97.32 % for accuracy and 99 % for detection rate, respectively. Forty features achieve an accuracy and detection rate of 99.4% and 99.3%, respectively. For 50 features, the accuracy and detection rate was 99.59 percent and 99.54 percent, respectively. For 60 features, the accuracy and detection rates were 99.6 percent and 99 percent, while for 78 features, they were 99.56 percent and 99.52 percent, respectively. Based on previous results, 60 features provided the highest accuracy, while 50 features provided the highest detection rate, the lowest false alarm rate, and the highest F1-score value. For the remainder of the experiments, testing was conducted with 50 features.

TABLE 7. Feature Selection based on binary UNSW-NB15.

UNSW-NB15 Binary classification (%)						
No. of feature	ACC	Pr	DR	F1-score	FAR	Training Time (s)
24	93.57	95.0	94.5	95.0	6.8	244
32	93.69	94.9	**94.90**	94.90	6.85	450
42	**93.70**	95.56	94.84	95.60	**6**	404

The results based on the binary UNSW-NB15 dataset are displayed in Table 7. For UNSW-NB15, we decided to utilize all 42 features from this dataset. Starting with one layer of CNN-LSTM, 24 features achieved an accuracy of 93.57 % and a detection rate of 94.5 %. For 32 features, the accuracy and detection rate was 93.69 and 94.80 %, while for 42 features, they were 93.7 and 94.84 %, respectively. The lowest FAR value among 42 features was 6. In addition, when we examined the training time, 42 features required less time to train the data than 32 features, so we continued testing with 42 features.

The final feature selection testing was based on binary WSN-DS. Eighteen features determined the optimal performance of a model. 18 features achieved 99.58 and 98.27 % accuracy and detection rate, compared to 88.89 and 97.04 % detection rate and 98.11 and 97.60 % accuracy for 12 and 6 features, respectively. In addition, the entire feature set was used for this dataset to train the IDS model.

We evaluated the Adam optimizer and RMSprop-based model. Adam optimizer was used to achieve the previous results. The accuracy and detection rates for one layer of CNN-LSTM based on RMSprop and CIC-IDS 2017 were 99.52 % and 99 % for CIC-IDS 2017, 93.57 % and 93 % for UNSW-NB15, and 99.60 % and 98.27 % based on WSN-DS, respectively. Therefore, we chose to continue using the Adam optimizer due to its superior accuracy and detection rate.

3) CNN-LSTM BASED ON THE NUMBER OF LAYERS AND HYPERPARAMETER

This section demonstrates the third portion of our testing, based on the number of layers, neurons, FC layers, and dropout layer rate.

The outcomes presented in Table 9 were very similar. The best performance was 99.6 % for three layers with a dropout rate of 0.2 and one FC layer, followed by 99.55 % for two layers with a dropout rate of 0.2 and two FC layers, and finally 99.56 % for one layer with a dropout rate of 0.2 and two FC layers. In order to select the structure with the

TABLE 8. Feature selection based on binary WSN-DS.

WSN-DS Binary classification (%)						
No. of features	ACC	Pr	DR	F1-score	FAR	Training Time (s)
6	97.60	80.5	97.04	97.6	2.6	57
12	98.11	90.59	88.89	98.8	5.9	84
18	**99.58**	97.35	**98.27**	98	**0.98**	112

TABLE 9. Structure selection based on the CIC-IDS2017 dataset.

CIC-IDS 2017 Binary classification					
No. of layers	No. of Neurons	Dropout rate	FC layer	Training accuracy %	Testing accuracy %
1	16	0.2	1	99.52	99.03
			2	99.44	**99.56**
		0.5	1	98.61	99.16
			2	99.2	99.46
2	16, 32	0.2	1	99.47	99.52
			2	99.38	**99.55**
		0.5	1	98.57	98.95
			2	99.2	99.53
3	16, 32, 64	0.2	1	99.50	**99.60**
			2	99.09	99.38
		0.5	1	98.74	98.68
			2	98.84	98.86
4	16, 32, 64, 128	0.2	1	99.11	99.45
			2	99.28	99.51
		0.5	1	98.53	99.03
			2	98.80	98.74

TABLE 10. Accuracies of selected structure of CIC-IDS2017 dataset.

CIC-IDS 2017 Binary classification (%)							
No. of Neuron	Train ACC	Train Loss	Test ACC	Test Loss	Valid. ACC	Valid. Loss	FAR
16	99.4	2.1	99.56	0.015	99.57	1.49	0.12
16, 32	99.4	2.2	99.55	0.015	99.54	1.6	0.122
16, 32, 64	99.5	1.2	99.60	0.013	99.60	1.2	0.11

highest performance, we analyzed additional data, as shown in Table 10. Table 10 indicates that the optimal structure consists of three layers with a dropout rate of 0.2 and one FC layer. The validation accuracy was identical to the testing accuracy at 99.60 %, and the FAR was the smallest at 0.11.

The same test was conducted on the binary UNSW-NB15 dataset, and the outcomes are presented in Table 11. According to the table, the three highest testing accuracies were 93.71 % for one layer with a 0.2 dropout rate and one FC layer, 93.65 % for one layer with a 0.2 dropout rate and two FC layers, and 93.63 % for one FC layer with 0.2 dropout rate and three CNN-LSTM layers. Based on the results presented in Table 12, we decided to continue training based on three layers of CNN-LSTM for UNSW-NB15. Despite not having the highest testing or training accuracy, the three structures had the highest validation accuracy with 93.7 %, the lowest loss with 11, and the lowest FAR with 6.2.

Based on WSN-DS, as shown in Table 13, the highest testing accuracy was achieved using one CNN-LSTM layer

TABLE 11. Structure selection based UNSW-NB15 dataset.

UNSW-NB15 Binary classification					
No. of layers	No. of Neurons	Dropout rate	FC layer	Training accuracy %	Testing accuracy %
1	16	0.2	1	93.31	93.62
			2	93.41	93.54
		0.5	1	92.96	93.34
			2	92.80	93.16
2	16, 32	0.2	1	93.48	**93.71**
			2	93.47	93.62
		0.5	1	93.04	93.35
			2	92.79	92.62
3	16, 32, 64	0.2	1	93.18	**93.63**
			2	93.51	**93.65**
		0.5	1	92.87	93.05
			2	92.32	92.81
4	16, 32, 64, 128	0.2	1	93.10	93.30
			2	93.39	93.57
		0.5	1	92.74	93.19
			2	92.25	92.90

TABLE 12. Accuracies of selected structure of UNSW-NB15 dataset.

UNSW-NB15 Binary classification (%)							
No. of Neuron	Train ACC	Train Loss	Test ACC	Test Loss	Valid. ACC	Valid. Loss	FAR
16, 32	93.5	12.7	93.71	12.7	93.66	12.7	6.3
16, 32, 64	93.5	13	93.65	13	93.60	12	6.5
16, 32, 64	93.2	12.9	93.63	0.13	93.70	11	6.2

TABLE 13. Structure selection based WSN-DS dataset.

UNSW-NB15 Binary classification					
No. of layers	No. of Neurons	Dropout rate	FC layer	Training accuracy %	Testing accuracy %
1	16	0.2	1	99.53	99.58
			2	99.52	99.47
		0.5	1	99.35	99.58
			2	99.21	**99.63**
2	16, 32	0.2	1	99.51	99.53
			2	99.46	**99.62**
		0.5	1	99.21	99.30
			2	98.99	99.27
3	16, 32, 64	0.2	1	99.60	**99.61**
			2	99.41	99.55
		0.5	1	99.12	99.41
			2	99.01	99.47
4	16, 32, 64, 128	0.2	1	99.53	99.32
			2	99.30	99.41
		0.5	1	99.09	99.13
			2	98.99	99.10

with a dropout rate of 0.5 and two FC layers. Then, 2 CNN-LSTM layers with 0.2 dropouts were followed by 2 FC

TABLE 14. Accuracies of selected structure of UNSW-NB15 dataset.

WSN-DS Binary classification (%)							
No. of Neuron	Train ACC	Train Loss	Test ACC	Test Loss	Valid. ACC	Valid. Loss	FAR
16	99.2	2.9	99.63	1.7	99.58	1.7	0.96
16, 32	99.5	2.12	99.62	1.6	99.58	1.69	1.05
16, 32, 64	99.6	2.1	99.61	1.9	99.60	1.9	0.90

layers. And finally, three CNN-LSTM layers with a dropout rate of 0.2 and one FC layer.

Comparing the highest three testing accuracy in terms of validation accuracy, loss, and FAR, we determined that the three layers had the highest validation accuracy at 99.60% and the lowest FAR at 0.90.

4) CNN-LSTM BASED ON STRATIFIED K-FOLD CROSS

After determining the optimal number of layers, neurons, FC layers, and dropout rate, we continued testing while varying the Stratified K-Fold cross parameter. The metrics for CIC-IDS based on binary and multiclass classification are displayed in Table 15. The highest accuracy was 99.64 % at $K = 8$ and $K = 4$, while the highest detection rate was 99.70 % for binary classification and 99.95 % for multiclass classification at $K = 8$. At $K = 8$ for binary classification and K=10 for multiclass classification, the smallest FAR achieved was 0.1. Tables also display the F1 score and precision values.

According to Table 16, there is a discernible distinction between binary and multiclass in the UNSW-NB15 dataset. At $K = 6$, the highest accuracy for binary classes was achieved with 93.95 %, compared to 82.2 % at $K = 4$ for multiclass. The highest detection rates achieved at $K = 8$ based on binary and multiclass classification were 94.53 and 82.41 %, respectively. For FAR, the lowest value was found at $K = 8$ for binary classes and $K = 4$ for multiclass, with a value of 2.2.

K had varying effects on the outcomes of the WSN-DS simulation. Based on Table 17, the best binary and multiclass accuracy were achieved at $K = 10$ with 99.67 % and 99.43 %, respectively. The highest detection rates occur at $K = 10$ and $K = 8$, with 98.14 % and 98.83 %, respectively. The lowest FAR achieved with $K = 6$ in binary and $K = 2$ in multiclass was 0.11 and 0.67, respectively.

The effect of modifying K-Fold on each record type in the datasets is depicted in the following figures.

Figures 5 and 6 illustrate the effect of training the model with CIC-IDS2017 data. Each type of record achieved a high detection rate and low FAR values, demonstrating robust implementation. There was a slight change in detection rates as K-Fold increased, but SFH-Patator had the most significant impact. For FAR, increasing K decreased the values of every record.

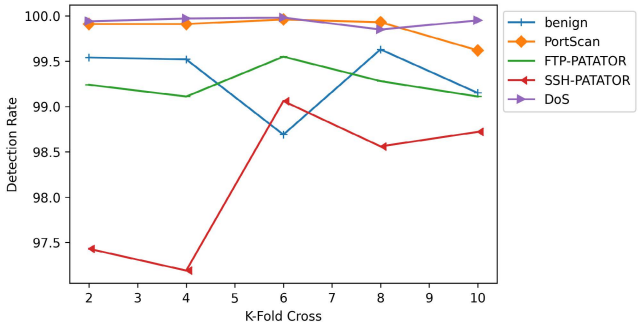

FIGURE 5. Effect of K-Fold cross on detection rate based on CIC-IDS2017 dataset.

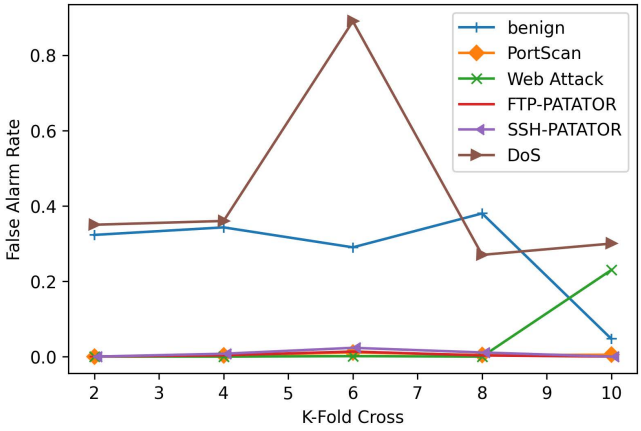

FIGURE 6. Effect of K-Fold cross on false alarm rate based on CIC-IDS2017 dataset.

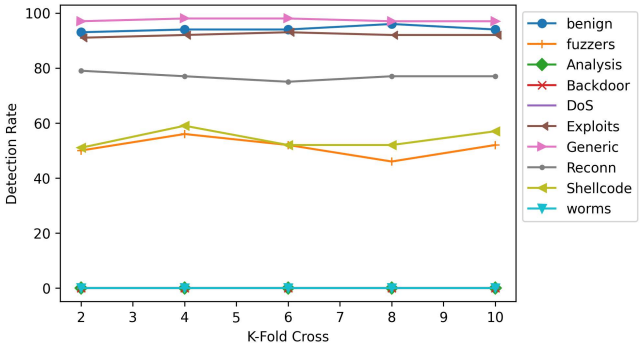

FIGURE 7. Effect of K-Fold cross on detection rate based on UNSW-NB15 dataset.

Figure 7 demonstrates that the detection rate for most record types at UNSW-NB is effective, particularly for records with a large number of records. Worm and DoS have the lowest detection rates, with detection rates approaching zero as K increases. The model classified these attacks as reconnaissance attacks based on the confusion matrix results. Figure 8 depicts K-fold versus FAR for the identical records, where DoS obtained the highest values of FAR.

Figure 9 and Figure 10 illustrate the WSN-DS performance. Increasing the number of K-Folds enhanced the performance of Blackhole attacks while decreasing the performance of

TABLE 15. Changing K-Fold cross based on CIC-IDS2017 dataset.

CIC-IDS 2017										
K	Binary classification (%)					Multiclass classification (%)				
	ACC	PR	DR	F1-score	FAR	ACC	PR	DR	F1-score	FAR
2	99.60	99.51	99.66	99.5	0.11	99.58	99.78	99.65	99.90	0.11
4	99.64	99.60	99.67	99.5	0.34	99.56	99.62	99.87	99.94	0.12
6	99.63	99.57	99.68	99.6	0.35	99.18	98.23	98.59	98.76	0.20
8	99.64	99.56	**99.70**	99.6	0.10	99.60	99.84	**99.95**	99.98	0.12
10	99.48	99.25	99.69	99.3	0.5	99.52	99.42	99.64	99.22	0.10

TABLE 16. Changing K-Fold cross based on UNSW-NB15 dataset.

UNSW-NB15										
K	Binary classification (%)					Multiclass classification (%)				
	ACC	PR	DR	F1-score	FAR	ACC	PR	DR	F1-score	FAR
2	93.63	93.44	93.56	93.20	6.2	81.20	80.46	81.47	78.33	2.34
4	93.93	93.56	93.89	94.00	6.35	82.20	82.12	82.33	80.43	2.22
6	93.95	93.22	93.46	93.31	6.45	82.0	81.69	81.99	80.69	2.3
8	93.78	94.69	**94.53**	94.77	6.0	81.83	81.59	**82.41**	80.87	2.3
10	93.59	93.89	93.67	93.88	6.45	81.99	82.69	82.12	80.33	2.3

TABLE 17. Changing K-Fold cross based on WSN-DS dataset.

WSN-DS dataset										
K	Binary classification (%)					Multiclass classification (%)				
	ACC	PR	DR	F1-score	FAR	ACC	PR	DR	F1-score	FAR
2	99.48	96.49	98.04	97	0.37	98.30	98	98	98	**0.67**
4	99.56	97.54	97.61	98	0.24	98.12	97.71	97.87	98.2	0.91
6	99.63	98.86	97.08	98	**0.11**	98.28	99	98.42	97.89	0.69
8	99.58	97.66	97.77	98	0.24	98.35	98.72	**98.83**	98.44	0.80
10	99.67	98.18	**98.14**	98	0.18	98.43	99.12	98.03	98.32	0.75

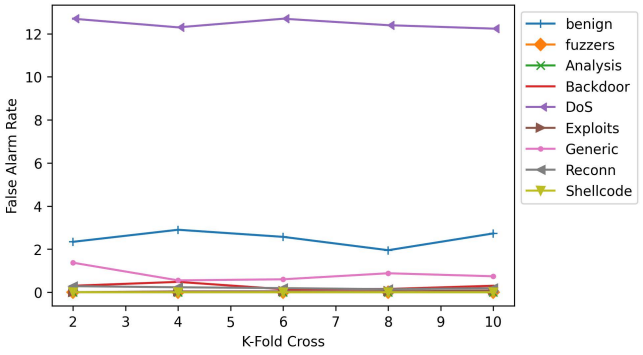

FIGURE 8. Effect of K-Fold cross on false alarm rate based on UNSW-NB15 dataset.

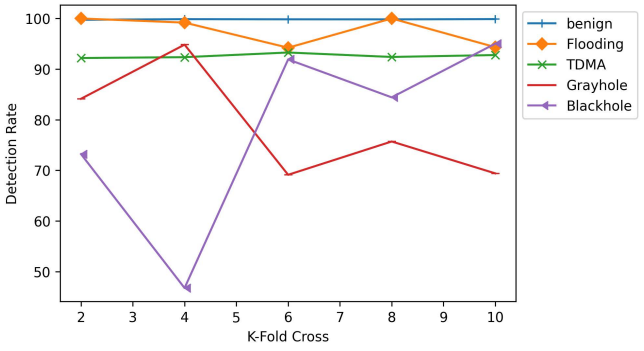

FIGURE 9. Effect of K-Fold cross on detection rate based on WSN-DS dataset.

Grayhole attacks. Similar detection rate values were observed for other records. Almost every K-Fold yielded poor results. We aim to enhance the model's ability to detect all attack types.

5) CNN-LSTM BASED ON EPOCH

After observing the impact of increasing the K-Fold cross, we examined the impact of increasing the number of epochs. Based on the previous values, we decided to continue testing with $K = 8$.

Figures 4.11 and 4.12 illustrate the effect of increasing the number of epochs on the binary classification detection rate and FAR. On UNSW-NB15, the number of epochs had the most significant impact, as the detection rate increased from 94.53 % at 5 epochs to 95.81 % at 60 epochs (refer to Figure 4.11). At 5 and 60 epochs, the accuracy of CIC-IDS

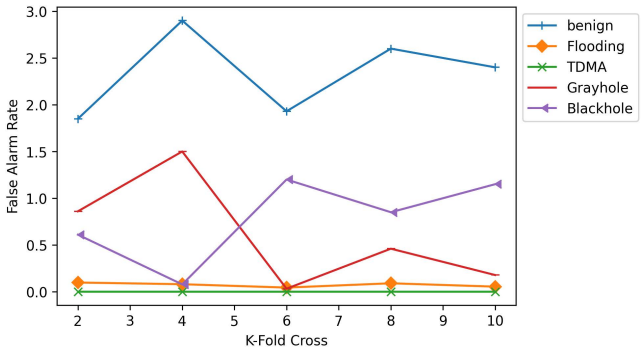

FIGURE 10. Effect of K-Fold cross on false alarm rate based on WSN-DS dataset.

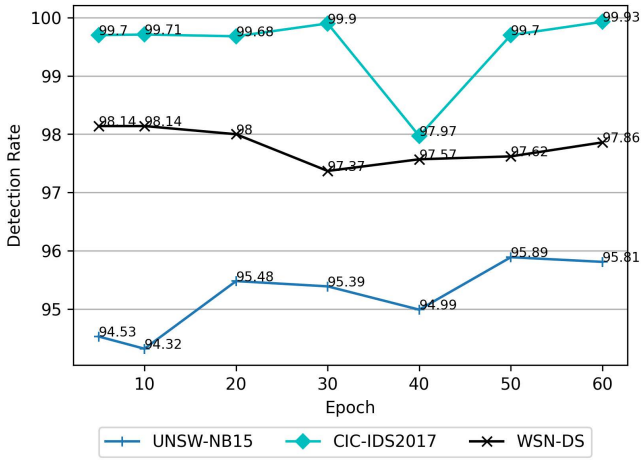

FIGURE 11. Effect of changing epoch on detection rate based on binary classification.

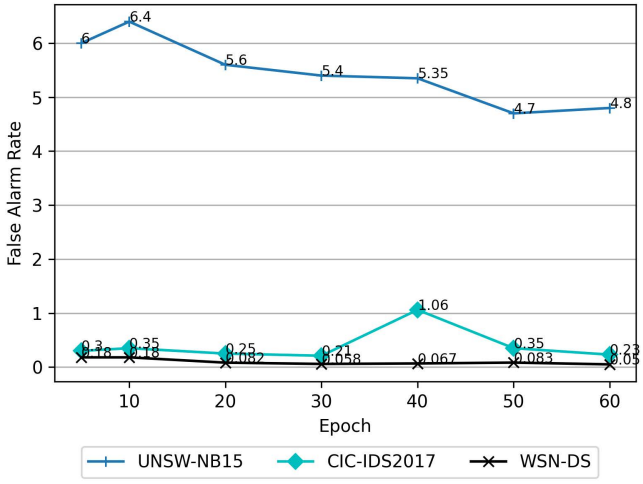

FIGURE 12. Effect of changing epoch on false alarm rate based on binary classification.

was 99.7 and 99.93 %, while that of WSN-DS was 98.14 and 97.86 %, respectively. Figure 4.12 demonstrates that the UNSW-NB15 dataset and other datasets obtained the highest FAR values.

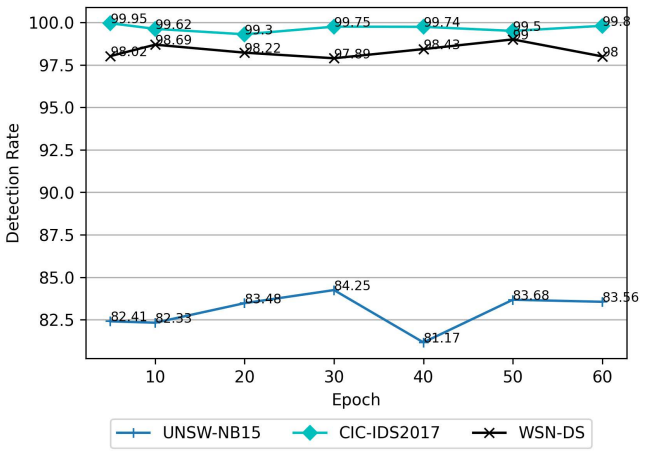

FIGURE 13. Effect of changing epoch on detection rate based on multiclass classification.

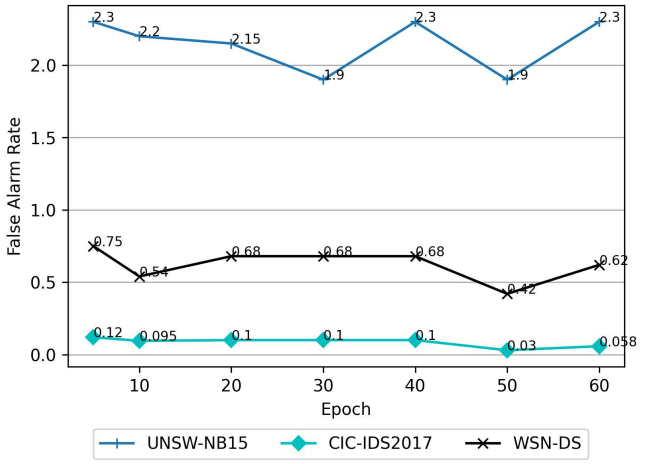

FIGURE 14. Effect of changing epoch on false alarm rate based on multiclass classification.

Figures 13 and 14 show that multiclass and binary classification performance is identical. UNSW-NB15 obtained the lowest detection rate values and the highest FAR values. As shown in the figures below, increasing epochs did not affect CIC-IDS2017 and WSN-DS.

The confusion matrices of the three datasets are shown in Figure 15. It demonstrates that the classification of the majority of record types was accurate, but PortScan attacks were predicted to be normal records. Figure 16 demonstrates that the most prevalent attack types were Exploits, Fuzzers, DoS, and worms, which the model classified as Reconnaissance attacks.

Due to the model's ability to accurately classify all types of records in the dataset, as depicted in Figure 17, the majority of records in each type were accurately predicted.

6) BENCHMARKING EVALUATION

As shown in the following tables, we compared the efficacy of our model to that of prior studies. The overall performance of our model surpasses that of other recent studies.

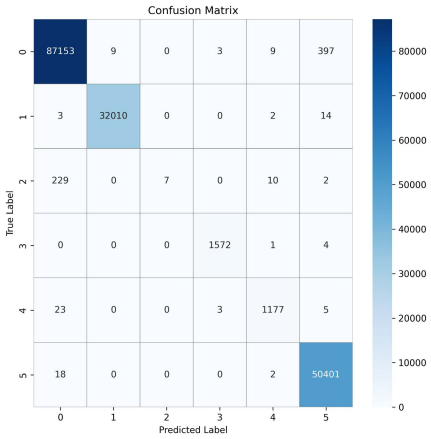

FIGURE 15. Confusion matrix for CIC-IDS2017 based on 5 epochs and K=8.

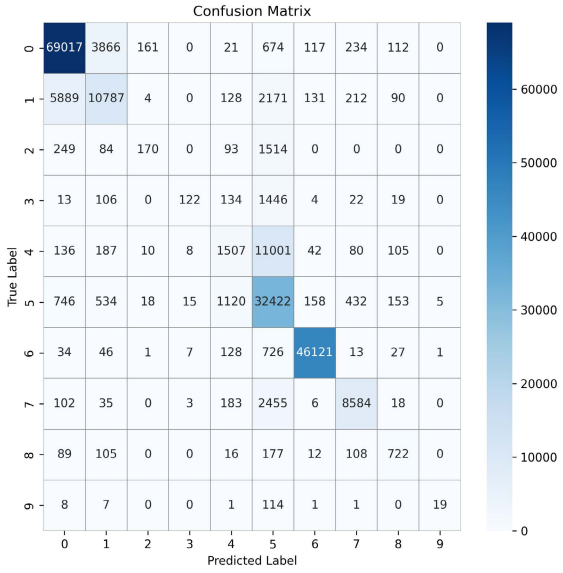

FIGURE 16. Confusion matrix for UNSW-NB15 based on 5 epoch and K=8.

Based on the dataset with 5 epochs, $K = 8$, and binary classification, we conducted a comparison. Beginning with UNSW-NB15, our model's performance exceeded other machine learning and deep learning structures. The accuracy of our CNN-LSTM-based model is 93.78 %, compared to 85.77 % for the Deep Belief Network (DBN) and 89.08 % for the Autoencoder with Deep Neural Network (ICVAE-DNN), and 82.42 % for the Support Vector Machine model (SVM). CNN-LSTM achieved the lowest values for FAR, with identical results. The detection rate yielded a slightly lower value than other models. However, the overall performance of our model was superior to that of other studies because we stacked CNN and LSTM layers, as shown in Table 18, based on UNSW-NB15.

The CIC-IDS2017 data set is utilized for another comparison. Table 19 demonstrates the robustness of our binary classification-based CIC-IDS2017 model. CNN-LSTM achieves 99.64 % accuracy, which is higher than Multilayer

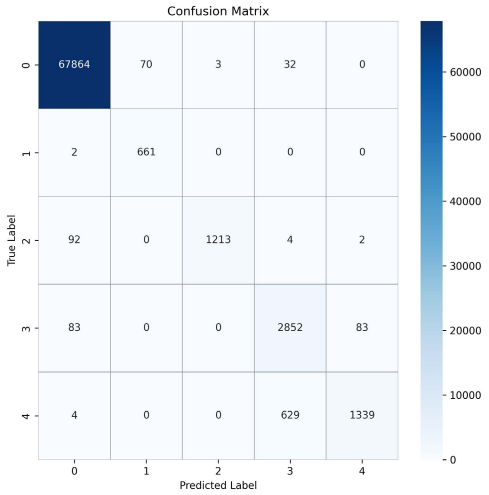

FIGURE 17. Confusion matrix for WSN-DS based on 5 epoch and K=8.

TABLE 18. Benchmarking based on the UNSW-NB15 dataset.

Author	Algorithm	ACC	FAR	DR
1	SVM	62.42	*	88.58
2	ICVAE-DNN	89.08	19.01	95.68
	DBN	85.77	30.32	**98.90**
Our work	CNN-LSTM	**93.78**	**6.0**	94.53

TABLE 19. Benchmarking based on the CIC-IDS2017 dataset.

Author	Algorithm	ACC	FAR	DR
[3]	KNN	80.91	*	91.28
[4]	REP Tree	96.67	1.145	94.47
[5]	MLP	85.24	7.35	77.83
Our work	CNN-LSTM	**99.64**	**0.10**	**99.70**

TABLE 20. Benchmarking based on the WSN-DS dataset.

Author	Algorithm	ACC	DR
[4]	LR	97.0	77.7
	NB	83.1	76.5
	DT	99.1	95.1
Our work	CNN-LSTM	**99.58**	**97.77**

Perceptron (MLP) with 85.24 % accuracy, Rep Tree with 96.67 % accuracy, and K-Nearest Neighbor (kNN) with 80.16 % accuracy. FAR and detection rate based on CNN-LSTM also produced superior results compared to other models.

Results in Table 20 show the performance base on the WSN-DS dataset. The accuracy achieved by our model was 99.58% outperforming other machine learning algorithms, whereas 97% was achieved by Logistic Regression (LR), 83.1% based on Naïve Bayes, and 99.1% based on Decision Tree (DT). Also, CNN-LSTM obtained the highest detection rate with 97.77%. Our results outperformed the benchmarked studies due to the structure of stacking layers of CNN and LSTM followed by DNN, cleaning the dataset,

choosing the best features, adding dropout, and adding batch normalization.

7) DISCUSSION

The primary objective of the research is to create an effective intrusion detection system that can distinguish between normal and malicious traffic. The number of new attacks discovered every day has increased the complexity of cybersecurity problems and traditional intrusion detection systems have a high false alarm rate, causing security analysts to ignore harmful attacks and leaving the system vulnerable to any type of attack. Data used to train intrusion systems is considered out of date and contains redundant information resulting in insufficient training and an ineffective training and evaluation process. Researchers have recently begun to develop intrusion detection systems based on deep learning. Recent research shows that deep learning outperforms conventional learning techniques in terms of detecting malicious traffic and classifying received traffic in massive data and continuous attacks. Many studies have used CNN and LSTM configurations, the difference is that they did so separately. In our model, we created a hybrid structure in which we combined the two algorithms, which means that in each step, CNN and LSTM will be used to process the data. In our study, we employed the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) algorithms (LSTM). Using three layers of hybrid CNN and LSTM, the structure of the model achieved our goal of delivering a model with a high detection rate and low accuracy. Preprocessing steps on datasets were performed, including encoding, normalizing data, and selecting the best features to train the model. The output was fed to the first layer of CNN to perform spatial feature extraction, then the LSTM layer to perform temporal feature extraction, and finally the FC layer to perform classification.

CIC-IDS2017 achieved the highest accuracies of 99.64 % for binary classification and 99.60 % for multiclass classification throughout 5 epochs. At the same time, the precision and F1- scores were 99.56 % and 99.6 % for binary classification and 99.84 % and 99.98 % for multiclass classification, respectively. Based on the binary and multiclass classification, the highest detection rate was achieved at $K = 8$ with 99.70% and 99.95%, and the lowest false alarm rate was achieved at 0.10% and 0.12%.

Based on 5 epochs of UNSW-NB15, the highest binary and multiclass detection rates at $K = 8$ were 94.53 and 82.41 %, respectively. The highest binary and multiclass classification accuracies achieved were 93.95 % at $K = 6$ and 82.20 % at $K = 4$, respectively. In contrast, the highest precision and F1-score for binary classes were 94.69 and 94.77 % at $K = 8$, whereas they were 82.69 and 80.87 % at $K = 10$ and $K = 8$, respectively, for binary and multiclass classification. The lowest false alarm rate for binary was 6 % at $K = 8$ and 2.22 % at $K = 4$.

Based on 5 epochs of binary WSN-DS at $K = 10$, the highest accuracy, detection rate, and F1-score were achieved: 99.67 %, 98.14 %, and 98 %, respectively. The highest precision and lowest false alarm rate were also achieved: 98.86 % and 0.11 %, respectively. On the other hand, Multiclass classification achieved the highest detection rate and F1-score at K = 8: 98.83 and 98.44 %, respectively, and the highest accuracy and precision at $K = 10$: 98.43 and 99.12 %, respectively. $K = 2$ had the lowest rate of false alarms, 0.67 %.

V. CONCLUSION AND FUTURE WORK

This study developed an intrusion detection system based on the CNN and LSTM deep learning algorithms. We stacked CNN and LSTM layers in our model and took advantage of CNN's ability to extract spatial features and LSTM's ability to extract temporal features. We implemented batch normalization, dropout layers, and standardization to improve our model. The model was evaluated using the UNSW-NB15, CIC-IDS2017, and WSN-DS datasets, all of which contained benign and attack records. As a first step, we tested the behavior of these datasets based on CNN, LSTM, CNN-LSTM, and LSTM-CNN. The results indicated that the CNN-LSTM hybrid model provided the highest detection rate and accuracy. Based on this, we evaluated the hybrid model based on binary and multiclass classification scenarios. With 5 epochs, we obtained 99.64 %, 94.53 %, and 99.67 % accuracy for binary classification using the CIC-IDS2017, UNSW-NB, and WSN-DS datasets, respectively. Although the model was unable to provide a high detection rate for certain types of attacks, such as web attacks in CIC-IDS2017 and worms, backdoors, and analysis in UNSW-NB15, the detection rate, and FAR results are encouraging. The effect of K-Fold cross-validation and increasing the number of epochs were examined, and the results indicated that the performance would initially improve before becoming stable. In the future, we intend to improve the model's performance in terms of its low detection rate and high FAR resulting from the dataset's imbalanced records.

REFERENCES

[1] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Secur. Commun. Netw.*, vol. 2020, Aug. 2020, Art. no. 8890306.
[2] M. Almansor and K. Gan, "Intrusion detection systems: Principles and perspectives," *J. Multidisciplinary Eng. Sci. Stud.*, vol. 4, no. 11, pp. 2458–2925, 2018.
[3] H. Alkahtani and T. H. H. Aldhyani, "Intrusion detection system to advance Internet of Things infrastructure-based deep learning algorithms," *Complexity*, vol. 2021, Jul. 2021, Art. no. 5579851.
[4] D. I. Edeh, "Network intrusion detection system using deep learning technique," M.S. thesis, Dept. Comput., Univ. Turku, Turku, Finland, 2021.
[5] P. Wu, "Deep learning for network intrusion detection: Attack recognition with computational intelligence," M.S. thesis, School Comput. Sci. Eng., Univ. New South Wales, Sydney NSW, Australia, 2020.
[6] M. K. Putchala, "Deep learning approach for intrusion detection system (IDS) in the Internet of Things (IoT) network using gated recurrent neural networks (GRU)," M.S. thesis, Dept. Comput. Sci. Eng., Wright State Univ., Dayton, OH, USA, 2017.
[7] H. Benmeziane, "Comparison of deep learning frameworks and compilers," M.S. thesis, Dept. Comput. Sci., École Nationale Supérieure d'Informatique, Oued Smar, Algeria, 2020.

[8] R. K. Vigneswaran, R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in *Proc. 9th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2018, pp. 1–6.
[9] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Dec. 2021, doi: 10.1186/s40537-021-00444-8.
[10] B. B. Rao and K. Swathi, "Fast kNN classifiers for network intrusion detection system," *Indian J. Sci. Technol.*, vol. 10, no. 14, pp. 1–10, 2017.
[11] W.-C. Shi and H.-M. Sun, "DeepBot: A time-based botnet detection with deep learning," *Soft Comput.*, vol. 24, pp. 16605–16616, May 2020.
[12] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine learning and deep learning approaches for cybersecuriy: A review," *IEEE Access*, vol. 10, pp. 19572–19585, 2022.
[13] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, Aug. 2016, Art. no. 4731953.
[14] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.
[15] R. Rustam, K. Ramli, N. Hayati, E. Ihsanto, T. S. Gunawan, and A. H. Halbouni, "Development of intrusion detection system using residual feedforward neural network algorithm," in *Proc. 4th Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI)*, Dec. 2021, pp. 539–543.
[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

ASMAA HALBOUNI (Member, IEEE) received the bachelor's degree in telecommunication engineering from An-Najah National University, Palestine. She is currently pursuing the M.S. degree in computer and information engineering with International Islamic University Malaysia, Malaysia. Her research interests include intrusion detection, network security, and deep learning.

TEDDY SURYA GUNAWAN (Senior Member, IEEE) received the B.Eng. degree *(cum laude)* in electrical engineering from Institut Teknologi Bandung (ITB), Indonesia, in 1998, the M.Eng. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2001, the Ph.D. degree from the School of Electrical Engineering and Telecommunications, The University of New South Wales, Australia, in 2007, and the Ir. degree from ITB, in 2022.

He was the Head of the Department of Electrical and Computer Engineering, from 2015 to 2016, and the Head of programme accreditation and quality assurance at the Faculty of Engineering, International Islamic University Malaysia, from 2017 to 2018. He has been a Professor since 2019. He has been a Chartered Engineer at IET, U.K., since 2016; a Insinyur Profesional Utama at PII, Indonesia, since 2021; a registered ASEAN Engineer, since 2018; and a ASEAN Chartered Professional Engineer, since 2020. His research interests include speech and audio processing, biomedical signal processing and instrumentation, image and video processing, and parallel computing. He was awarded the Best Researcher Award, in 2018 from IIUM. He was the Chairperson of IEEE Instrumentation and Measurement Society—Malaysia Section, in 2013, 2014, 2021, and 2022.

MOHAMED HADI HABAEBI (Senior Member, IEEE) is currently a Professor with the Department of Electrical and Computer Engineering, International Islamic University Malaysia (IIUM). His research interests include the IoT, mobile app. development, networking, blockchain, AI applications in image processing, cyber-physical security, wireless communications, small antennas, and channel propagation modeling.

MURAD HALBOUNI graduated the bachelor's degree in telecommunication engineering from Palestine Technical University in Kadoorie, Palestine. He is currently pursuing the M.S. degree in cyber crime with Arab American University, Palestine. He works at Paltel, a Palestinian Communication Business, as a Networking Engineer. His research interests include cybercrime and digital evidence analysis, metro networks, network security, and machine learning.

MIRA KARTIWI (Member, IEEE) is a Professor with the Department of Information Systems, Kulliyyah of Information and Communication Technology, and also the Deputy Director of e-learning at the Centre for Professional Development, International Islamic University Malaysia (IIUM). She is also an experienced consultant specializing in the health, financial, and manufacturing sectors. Her research interests include health informatics, e-commerce, data mining, information systems strategy, business process improvement, product development, marketing, delivery strategy, workshop facilitation, training, and communications. She was the recipient of the Australia Postgraduate Award (APA), in 2004. For her achievement in research, she was awarded the Higher Degree Research Award for Excellence, in 2007. She has also been appointed as an Editorial Board Member in local and international journals to acknowledge her expertise.

ROBIAH AHMAD (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Evansville, Evansville, IN, USA, the M.Sc. degree in information technology for manufacture from the Warwick Manufacturing Group, University of Warwick, U.K., and the Ph.D. degree in mechanical engineering from University Teknologi Malaysia, Malaysia. She is currently an Associate Professor with the Razak Faculty of Technology and Informatics, UTM, Kuala Lumpur, Malaysia. She has more than 20 years experience as a research scientist. She has published more than 100 peer-reviewed international journal articles/proceedings in the areas of instrumentation and control, system modeling and identification, and evolutionary computation. She is also an Executive Committee for Humanitarian Activities for IEEE Malaysia Section and the Past Chair for IEEE Instrumentation and Measurement Society Malaysia Chapter.