Article

Not peer-reviewed version

Hybrid Deep Neural Network-Based Detection of DDoS Attacks in Software-Defined IIoT Networks

Xiaoan Bao , Enlai Chen , Xiaomei Tu *

Posted Date: 13 January 2025

doi: 10.20944/preprints202501.0925.v1

Keywords: Hybrid Deep Neural Networks; Distributed Denial-of-Service(DDoS) attack detection; Adaptive Feature Selection Boosting(AFSB); Kolmogorov–Arnold Networks(KAN); Industrial Internet of Things(IIoT); software defined networking(SDN)

Article

Hybrid Deep Neural Network-Based Detection of DDoS Attacks in Software-Defined IIoT Networks

Xiaoan Bao [1], Enlai Chen [2] and Xiaomei Tu *

[1] College of Computer Science and Technology (College of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou 310018, China
[2] College of Computer Science and Technology (College of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou 310018, China
* Correspondence: txm_95@163.com

Abstract: In the current landscape of the Industrial Internet of Things (IIoT), Distributed Denial of Service (DDoS) attacks represent a significant security threat. Traditional defense mechanisms often require extensive computational and storage resources, resulting in substantial increases in operational costs. In response to this challenge, this study proposes a novel DDoS attack detection method for IIoT environments, named IIoT Attack Detection based on CNN-mLSTM-KAN (IAD-CLK). The method first employs an Adaptive Feature Selection Boosting (AFSB) technique during the data preprocessing phase to identify the most relevant features, thereby reducing the computational load on the model. Subsequently, the CNN-mLSTM-KAN model is introduced, which combines depthwise separable convolutions, an LSTM architecture enhanced by matrix operations (mLSTM), and the Kolmogorov–Arnold Network (KAN). This integration significantly improves the efficiency and accuracy of DDoS attack detection. Experimental results on the CICDDoS2019 dataset demonstrate that the model achieves an accuracy of 99.78%, while maintaining a low time cost of 0.122 ms. These findings not only highlight the model's advantages in terms of accuracy and computational complexity but also demonstrate its ability to meet the stringent low-latency requirements of IIoT systems.

Keywords: hybrid deep neural networks; Distributed Denial-of-Service(DDoS) attack detection; Adaptive Feature Selection Boosting(AFSB); Kolmogorov–Arnold Networks(KAN); Industrial Internet of Things(IIoT); software defined networking(SDN)

1. Introduction

The Industrial Internet of Things (IIoT) establishes a substantial ecosystem comprising sensors, actuator connections, and robust communication and control systems. It has demonstrated significant capabilities across various domains, including smart cities, manufacturing, healthcare, agriculture, and supply chain management. The extensive adoption of heterogeneous communication facilitates enterprise networks' connection through the global Internet, endowing robots with intelligence and enabling real-time data analysis for autonomous operations and precise decision-making [1,2]. Within this framework, Software-Defined Networking (SDN) serves as the foundational platform for IIoT [3], leveraging OpenFlow protocols and IEEE 802.1 time-sensitive networking to ensure dependable, flexible, and efficient network management and traffic provisioning on platforms necessitating low latency and real-time responses [4–6].

However, as IIoT systems transmit considerable volumes of sensitive data to the cloud, they encounter escalating network security threats. Of particular concern is the risk posed by Distributed Denial-of-Service (DDoS) attacks targeting network and application layers. These attacks exploit vulnerabilities within the network, transport, and application layers [7,8], including flooding techniques utilizing Synchronization (SYN) and User Datagram Protocol (UDP) [9,10]. Such assaults can severely disrupt the functionality of targeted servers. In order to mitigate these risks and safeguard the integrity, privacy, and security of IIoT networks, this research explores strategies for detecting

 doi:10.20944/preprints202501.0925.v1

DDoS flooding attacks within IoT frameworks. Despite advancements in identifying DDoS flooding attacks through statistical, knowledge-based methods, and machine learning (ML), their efficacy in real-time environments remains constrained [11,12]. While ML-based DDoS detection demonstrates high accuracy, its performance tends to diminish as the number of features increases, necessitating a separate feature extraction phase when applied to unlabeled datasets. To tackle these challenges, this study adopts deep learning (DL) techniques for the detection and classification of DDoS attacks [13]. Given the imperative for advanced DDoS detection with low latency in industrial IoT contexts, this research presents the following contributions:

(1) We have employed a variety of machine learning techniques and analyzed feature selection methods using performance metrics such as precision, accuracy, F1 score, recall, loss, and time cost. Based on these indicators, we selected the AFSB algorithm to reduce feature dimensionality. This feature selection strategy effectively identifies significant features, thereby enhancing performance robustness.

(2) We propose a hybrid approach for detecting IoT Distributed Denial-of-Service (DDoS) attacks, termed IAD-CLK, which combines convolutional neural networks (CNN) with an enhanced long short-term memory network utilizing matrix operations (mLSTM). The IAD-CLK method integrates a CNN-mLSTM-KAN model that employs a deeply separable convolutional neural network alongside mLSTM [14] and KAN [15] neural networks. This proposed model efficiently learns the intrinsic features of DDoS attacks in a computationally effective manner. Furthermore, residual connections are employed to enhance learning efficiency and mitigate the gradient vanishing problem. Experimental results demonstrate that this method achieves high detection accuracy and rapid convergence, facilitating effective detection of DDoS attacks in industrial Internet environments.

The remainder of this paper is organized as follows: Section II outlines the current state-of-the-art in the proposed DDoS attack detection model. Section III presents the experimental results and ensuing discussion. Finally, Section IV summarizes the research and explores future directions for this study.

2. Related Work

In general, two primary techniques are employed for the detection of Distributed Denial of Service (DDoS) attacks: 1) Machine Learning (ML) approaches and 2) Deep Learning (DL) approaches [16]. However, the existing methods exhibit significant limitations in meeting the low-latency requirements necessary for real-time DDoS detection in industrial Internet of Things (IoT) environments. Notably, feature engineering plays a crucial role in processing data generated from various applications. This section summarizes the state of the art in the aforementioned approaches and outlines their associated challenges.

2.1. ML Approaches

Machine Learning (ML) techniques have demonstrated significant potential in addressing detection challenges [17]. For instance, Ahakonye [18] examined the efficacy of detecting and classifying encrypted Supervisory Control and Data Acquisition (SCADA) systems using machine learning classifiers in a smart factory setting. Another study [19] investigated a low-cost, machine learning-based approach for detecting and classifying DDoS flooding attacks within Software Defined Networking (SDN) architecture. In web applications, both TCP and UDP generate the most substantial traffic, which is leveraged by the proposed intrusion detection system. Popular machine learning techniques for detecting DDoS attacks include Gaussian Naive Bayes (GNB), Classification and Regression Trees (CART), and Quadratic Discriminant Analysis. Among these methods, the CART technique achieves an accuracy of 98% in experiments and demonstrates strong performance. However, the computation time for these methods can reach up to 12.4 ms, which is relatively high for industrial Internet of Things (IoT) applications that require real-time operations [19].

 doi:10.20944/preprints202501.0925.v1

2.2. DL Approaches

In response to Distributed Denial of Service (DDoS) attacks, researchers have proposed a range of deep learning (DL) methods recognized for their robustness and ability to learn and identify significant features from network traffic [20–22]. For instance, Wei [13] developed a hybrid deep learning self-encoder multilayer perceptron (MLP) network that incorporates automatic feature extraction. This system achieves an accuracy of 98.34% on the CICDDoS2019 dataset by extracting essential features in scenarios involving compression and feature reduction via an autoencoder. Other studies have also analyzed this dataset [10,23]. In contrast, Amaizu [9] utilized a composite deep learning model to detect DDoS attacks in B5G networks, integrating two deep neural networks with distinct architectures. However, the deep neural network implemented in this model consists of seven layers, resulting in relatively high computational time.

Experimental methods for Intrusion Detection Systems (IDS) are frequently employed, as demonstrated by Yungaicela-Naula [10], who established a network based on a modular Software-Defined Networking (SDN) architecture. They targeted DDoS attacks at both the application and transport layers by employing machine learning (ML) and deep learning (DL) models. Their validation occurred in a simulated SDN environment, utilizing the Mininet simulator alongside the Open Network Operating System as the SDN controller. The most recent CICDDoS2017 and CICDDoS2019 security datasets were utilized in their study. The experimental setup aligns closely with the system design proposed in our research.

The evaluated system targeted DDoS attacks at the transport layer, including UDP and SYN floods, while also considering application layer DDoS attacks. The study encompassed various machine learning and deep learning models, including k-Nearest Neighbors (KNN), Random Forest (RF), Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), Multilayer Perceptron Networks (MLP), and Gated Recurrent Units (GRU). In the experiments, the models exhibited detection rates of 94.5% for slow attacks and 97.5% for high-volume attacks. Sahu [24] developed a hybrid approach that combines LSTM networks with Fully Connected Networks to classify benign and malicious network traffic through hyperparameter optimization. This approach addressed the imbalanced distribution of intrusion data between majority and minority classes, assessing binary and multiclass intrusion detection scenarios by comparing six network security datasets. Nonetheless, existing ML and DL methods [10,25,26] face challenges, including high computational complexity and inadequate efficiency in processing advanced DDoS attack models.

This research aims to present an effective Intrusion Detection System (IDS) technique specifically designed for Industrial Internet of Things (IIoT) networks, prioritizing low latency and reduced complexity. It employs a network monitoring system integrated with a Software-Defined Networking (SDN) controller to continuously capture network traffic from the IIoT, encompassing both benign and malicious flows. The study considers various types of Distributed Denial of Service (DDoS) attacks that utilize reflection and exploitation methods. Network devices, such as routers and switches (as illustrated in Figure 1), collect all incoming and outgoing traffic. Traffic characteristics are derived from datasets obtained using CICFlowMeter [27], and a feature selection mechanism is employed to enhance the accuracy of the proposed model.

 doi:10.20944/preprints202501.0925.v1

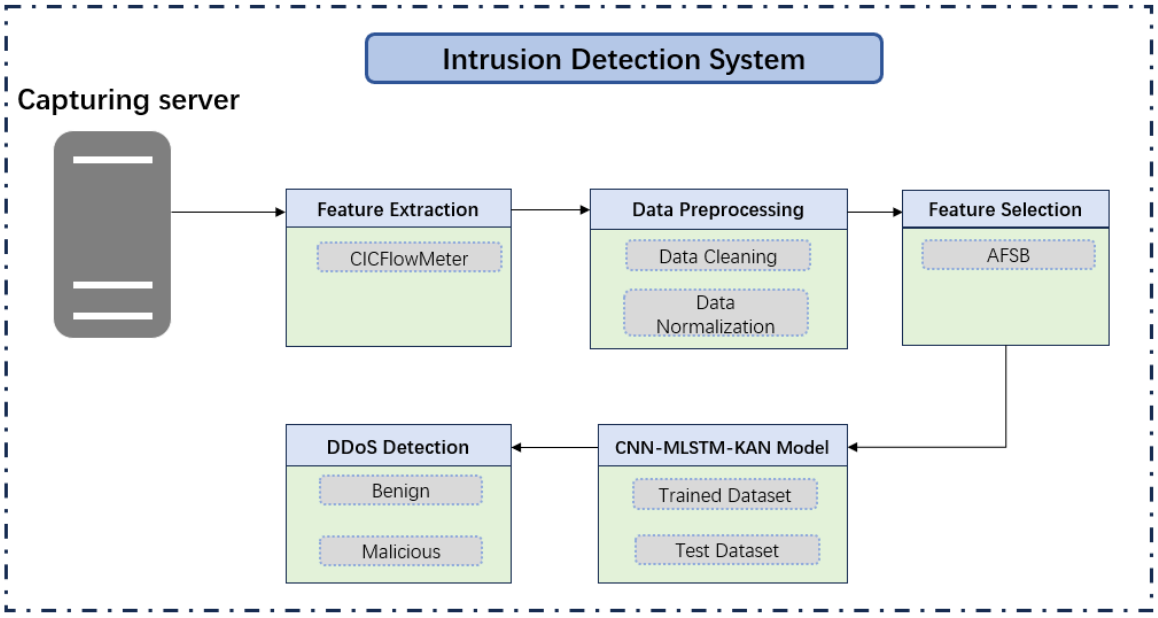

Figure 1. Flow Capture Mechanism Diagram.

3. Proposed Approach

This study introduces a novel technique for detecting and classifying Distributed Denial of Service (DDoS) attacks within the framework of an SDN-based Industrial Internet of Things (IIoT) network. The SDN controller serves as the central component for managing the network, facilitating all communications between applications and network devices. The northbound interface enables the controller to interact with applications for various functions, including network monitoring, flow management, network administration, load balancing, and Network Intrusion Detection Systems (IDS). Meanwhile, the southbound interface, utilizing protocols such as OpenFlow, allows communication between the controller and specific devices within the data plane. By leveraging these southbound protocols, the controller can orchestrate network devices and identify the most efficient paths for application traffic. Given the diverse range of responsibilities overseen by the controller, a reliable and low-complexity IDS is essential for effective attack detection. This paper proposes a DDoS detection method based on the CNN-mLSTM-KAN model, specifically tailored for IIoT network environments.

The workflow implemented in this research is illustrated in Figure 1. Initially, a capture server employs the CICFlowMeter tool [28] to collect network traffic from the core switch and extract 88 raw traffic features from PCAP files. Prior to feeding the data into the proposed model, preprocessing steps are necessary to ensure data integrity, including data cleaning, normalization, and feature selection. The proposed model operates within the SDN controller, employing various feature selection techniques to identify the most relevant features that significantly enhance DDoS attack detection performance. This feature selection process incorporates methods such AFSB to ascertain the most effective features. The selected features are subsequently input into the proposed model for the detection of DDoS attacks.

3.1. Data Preprocessing

Before the data are fed into the proposed model, preprocessing is necessary to ensure data quality. This step involves several processes, including data preparation and normalization.

(1) Data Preparation: The initial flow-based dataset contained 88 features. Several non-contributory features, such as "Unnamed," "Timestamp," "Source Port," "Source IP," "Flow ID," and "Similar HTTP," were removed. After discarding these features, 80 features remained for subsequent analysis. It was also essential to eliminate values containing NaN, infinite values, and empty entries. Given the large volume of samples in the dataset, instances with malformed values were removed.

 doi:10.20944/preprints202501.0925.v1

(2) Data Normalization: Some features, including "Bwd IAT min," "Flow IAT Std," and "Flow IAT Max," "Flow Duration," displayed considerable variance between their minimum and maximum values. To address this high variability among features, data normalization was applied. This technique not only shortens the training duration but also improves the performance of the model. In this research, feature scaling was implemented using a Min-Max normalization method, which is based on the following principles:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

In the formula, X_{sc} represents the normalized numerical result, which ranges between [0, 1]. The terms 'max' and 'min' denote the maximum and minimum values of the represented feature, respectively. The performance of DDoS detection models declines as the dimensionality of features increases. Some features exhibit little or no relevance to the DDoS detection process. To enhance the model's capability and reduce training time, it is essential to eliminate redundant datasets by decreasing feature dimensions. We employed the AFSB method to select the top 10 features with the highest weights. Table 1 presents the ten extracted features along with their corresponding values.

Table 1. Feature extraction Table.

Feature	Sample
Time	0.092
PKT Len	151
IP Flags	0X4000
Highest Layer	99602525
Protocols	0011010001006
TCP ACK	336
TCP Len	85
TCP Window Size	144
TCP Flags	0X018
UDP Len	0

AFSB is a novel algorithm that integrates feature selection with reinforcement learning, aimed at optimizing the performance of classifiers, particularly for datasets with high-dimensional feature spaces. AFSB initially evaluates the contribution of each feature to model performance through an adaptive mechanism, then dynamically adjusts the feature subset to ensure only the most critical features are used to enhance model prediction accuracy. The core innovation of this method lies in its continuous assessment and selection of features throughout the entire model training process, allowing the model to better adapt to the complex patterns and changes in the data. Additionally, AFSB introduces an optimization algorithm that adjusts the weights of features based on prediction errors in each round of reinforcement learning, further enhancing the specificity and efficiency of feature selection.

3.2. CNN-mLSTM-KAN

3.2.1. KAN

In the current field of deep learning, multilayer perceptual machines (MLPs) have been widely used for various tasks due to their simplicity and effectiveness. However, MLPs suffer from some fundamental problems such as vanishing and exploding gradients, parameter inefficiency and limited ability to handle high-dimensional data. To address these challenges, the model in this paper introduces a novel neural network structure, Kolmogorov-Arnold Networks (KAN), which is inspired by the Kolmogorov-Arnold Representation Theorem.KAN networks do not use the traditional Instead of using the linear weights and fixed activation functions of traditional MLP, KAN network adopts a

 doi:10.20944/preprints202501.0925.v1

nonlinear spline function (SPLIN function) as the dynamic activation function, which is an innovation that makes KAN structurally more flexible and efficient.KAN replaces the traditional combinations of weights and activation functions by a learnable activation function on each connection, which completely eliminates the linear weights. This strategy not only reduces the number of parameters of the model, but also greatly improves the ability of the model to handle complex functional relationships. In practical applications, KAN can reach or exceed the performance level of larger MLPs with a smaller network structure, especially in the function fitting task. In addition, by adjusting the refinement of the spline function grid, KANs can gradually improve the accuracy without retraining the whole model, and this flexibility is extremely useful in practice.

3.2.2. mLSTM

mLSTM (Matrix Long Short-Term Memory) enhances the memory capacity and parallel processing capabilities of the model by extending the vector operations of traditional Long Short-Term Memory networks to matrix operations.(mLSTM cell as illustrated in Figure 2) In mLSTM, each state is represented as a matrix instead of a single vector, enabling the capture of more complex data relationships and patterns within a single time step. mLSTM is particularly well-suited for tasks involving large-scale datasets or recognizing highly intricate data patterns. Furthermore, the design of mLSTM facilitates high levels of parallelization, which not only improves computational efficiency but also enables the model to scale effectively with large datasets.

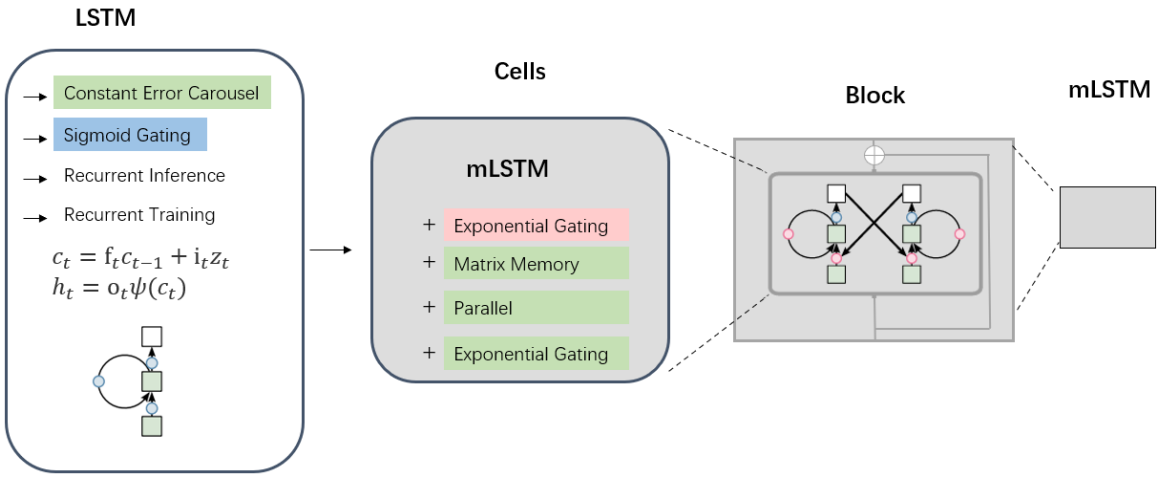

Figure 2. mLSTM memory cells.

This section presents the proposed hybrid Deep Neural Network (DNN) model, which integrates Convolutional Neural Network (CNN) and an LSTM architecture enhanced by matrix operations (mLSTM) layers, along with a KAN layer. This model addresses the challenges of accuracy and computational complexity associated with existing DDoS detection methods in SDN-based industrial Internet of Things (IoT) networks. Numerous studies have demonstrated that an increase in network size and computational resources typically enhances detection accuracy. However, significant concerns regarding computational costs and the limited number of parameters persist in detecting DDoS attacks within industrial IoT environments. To address these challenges, this research explores the implementation of depthwise separable convolutions and residual connections, optimizing parameter aggregation to improve learning efficiency while reducing computational expenditures. As illustrated in Figure 3, the proposed model employs depthwise separable convolutions instead of standard convolutional layers, utilizes mLSTM layers for flow identification, and replaces traditional Multi-Layer Perceptron (MLP) layers with a KAN at the output stage.

 doi:10.20944/preprints202501.0925.v1

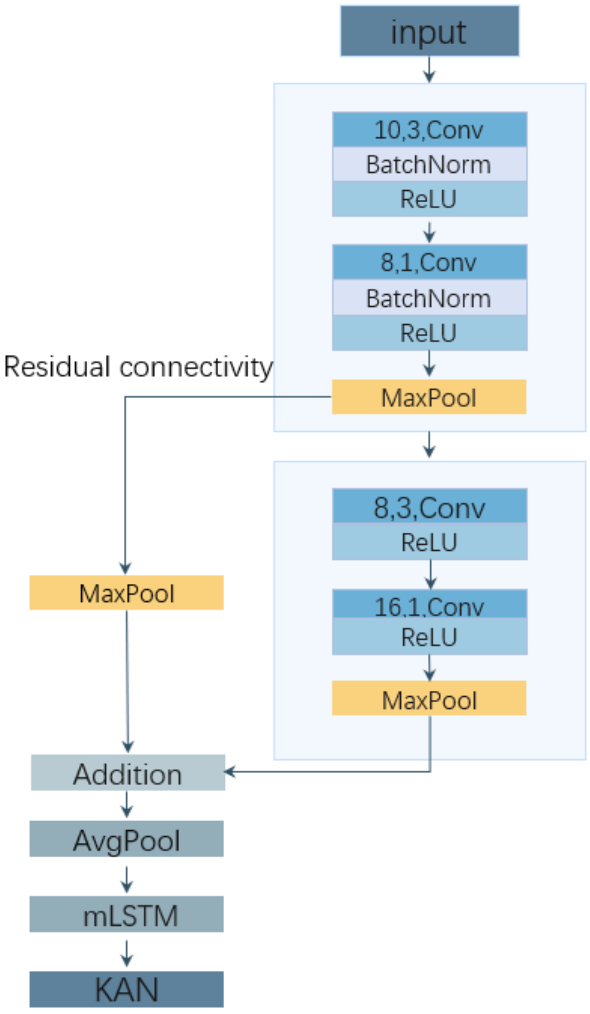

Figure 3. CNN-mLSTM-KAN-base proposed model.

Initially, input data, processed using previously introduced data preprocessing methods, enters the model, where the input layer is configured as $1 \in \mathbb{N}^{\text{batchwise} \times 10 \times 1}$. The model then proceeds to its first part, which consists of two depth wise separable one-dimensional convolutional layers. This type of convolutional layer first applies a depth wise convolution to independently filter each input features. one filter per input channel, followed by a pointwise convolution (1x1 convolution) to combine the output channels of the depth wise convolution. This configuration not only reduces the model's parameters but also retains the effective information of features. After the first convolutional layer, where the number of input channels equals the number of features, a convolution kernel of size three is used with an equal number of output channels to preserve all spatial relationships of the input features. A batch normalization layer is added post-convolution to accelerate the training process and enhance the model's generalization ability. To prevent the loss of useful feature information during convolution due to input feature shape, zero-padding is employed where dimensions are insufficient, ensuring that the output shape is correctly passed to the next layer. Subsequently, a ReLU activation function is applied, followed by a max pooling layer to reduce feature dimensions and increase the model's abstraction capabilities.

$$g(z) = \begin{cases} z, & \text{if} z \geq 0 \\ 0, & \text{if} z < 0. \end{cases} \tag{2}$$

The second convolutional layer also adopts the strategy of depth wise separable convolutions but increases the number of output channels from 8 to 16 to enhance the model's representational capacity. In this layer, in addition to the standard convolution, ReLU activation, and pooling operations, a

 doi:10.20944/preprints202501.0925.v1

residual connection is introduced. The residual connection involves adding an extra 1x1 convolution and a pooling layer to directly add the output of the first convolutional layer to the output of the second convolutional layer. This approach helps to alleviate the issue of vanishing gradients during the training of deep networks while enhancing the efficiency of information transfer.

In the subsequent phase of the model, a Matrix Long Short-Term Memory (mLSTM) layer is incorporated to process the output sequence generated by the convolutional layers. This mLSTM layer is configured with an input size of 16 and a hidden size of 32, comprising two stacked layers. Such a configuration allows the layer to effectively manage long-term dependencies in sequential data, thereby enhancing the model's performance in time-series analysis.

To conclude, a KAN layer is implemented to generate the model's final output. The output from the mLSTM layer is directed into the KAN layer, which serves to identify DDoS attacks across each data stream. The comprehensive network architecture of the proposed model is presented in Table2.

Table 2. Framework of the Proposed Network Design.

Component	Output Size	Detail
Input	[batchsize,10,10]	
Conv	[batchsize,10,10]	10,3,ReLU,padding
Batch Normalization	[batchsize,10,10]	
Conv	[batchsize,8,10]	8,1,ReLU,padding
Batch Normalization	[batchsize,8,10]	
MaxPool	[batchsize,8,5]	Pooling size = 2
Conv	[batchsize,8,5]	8,3,ReLU,padding
Conv	[batchsize,16,5]	16,1,ReLU,padding
MaxPool	[batchsize,16,2]	Pooling size = 2
Conv	[batchsize,16,5]	16,1,ReLU,padding
MaxPool	[batchsize,16,2]	Pooling size = 2
Addition	[batchsize,16,2]	
AvgPool	[batchsize,8,2]	Pooling size = 2
mLSTM	[batchsize,16,2]	
KAN	[batchsize,2]	

4. Experimental Findings and Interpretation

4.1. Experimental Environment

Table 3 outlines the optimal parameter configurations for the proposed model. The model exhibits impressive accuracy with the following training setup: a selection of ten features, 50 epochs of training conducted with a small batch size of 32 samples, an initial learning rate of 0.001, ReLU as the chosen activation function, the Adam optimizer combined with a cross-entropy loss function, and the application of 5-fold cross-validation. Training was carried out on two NVIDIA GeForce RTX 3090 GPUs within a 64-bit Ubuntu environment, utilizing PyTorch as the deep learning framework.

Table 3. Best Parameter Settings for the Suggested Model.

Param name	Param value
Selected Features	10
Total Features	80
Learning rate	0.001
Optimizer	Adam
Loss Function	Cross-entropy loss
Batch Size	32
Epoch	50
Activation Function	ReLU

 doi:10.20944/preprints202501.0925.v1

4.2. Experimental Dataset

Intrusion Detection System (IDS) evaluations typically utilize several publicly available datasets, such as NSL-KDD, DARPA/KDD99, and CAIDA [29]. These datasets have been recommended for Software-Defined Networking (SDN) architectural scenarios [30,31] and have been employed by A. Aldweesh [32] and Y. N. Kunang [33]; however, they do not encompass recent DDoS attack detection. Some researchers [10] have assessed their SDN-based DDoS detection techniques using the recent CICDDoS2019 dataset. This dataset comprises 88 traffic features relevant to SDN networks, facilitating the straightforward processing and extraction of complex features via SDN controllers. The current study utilizes the CICDDoS2019 dataset [27], which includes samples of recent DDoS attacks alongside benign traffic provided by the Canadian Institute for Cybersecurity Research. The CICDDoS2019 dataset offers a diverse collection of network traffic samples and features, surpassing the richness of existing datasets. It contains raw network traffic data in PCAP format, along with flow-based features derived using CICFlowMeter and saved in CSV format. The flow-based dataset encompasses 88 features and identifies 12 different types of DDoS attacks, including Simple Service Discovery Protocol (SSDP), UDP, UDP-Lag, SYN, Microsoft Structured Query Language (MSSQL), DNS, Temporary File Transfer Protocol (TFTP), NetBIOS, Simple Network Management Protocol (SNMP), WebDDoS, and NTP.

The CICDDoS2019 dataset includes various types of DDoS attacks based on reflection and exploitation methods. Reflection-based DDoS attacks use reflection servers to send packets that conceal the attack's source by utilizing the IP address of the intended victim. These attacks typically employ protocols such as TCP, UDP, or a combination of both; examples include SSDP and MSSQL attacks for TCP, as well as NTP and TFTP attacks for UDP. Additionally, these attacks can target services such as NetBIOS, LDAP, and DNS. Specifically, DNS flooding attacks disrupt domain name resolution, leading to degraded website responsiveness.

(1) Exploitation-based DDoS attacks: These attacks utilize TCP SYN flooding and UDP flooding techniques to deplete the victim's resources and network bandwidth. In a TCP SYN flood attack, an overwhelming number of SYN packets are sent, causing the server to become unresponsive to connection requests as it is unable to process the replies. Meanwhile, a UDP flood attack sends a significant volume of UDP packets to occupy the ports of the targeted system, draining the network bandwidth and potentially resulting in a system crash.

(2) Exploitation-based DDoS attacks: These attacks leverage TCP SYN flooding and UDP flooding to overload the victim's resources and network capacity. In TCP SYN flooding, a multitude of SYN packets is transmitted, preventing the server from handling new connection requests due to its inability to respond to incoming replies. Conversely, UDP flooding attacks target the ports of the victim's system with numerous UDP packets, which can exhaust network bandwidth and ultimately lead to a system failure.

Since most of the traffic in the CIC-DDoS2019 dataset belongs to attack traffic, and there is very little normal traffic, we imported the attack traffic in the CIC-DDoS2019 dataset, and at the same time, and we also externally selected some of the normal traffic in *.pcap format obtained from the packet capture.

4.3. Experimentation and Verification

4.3.1. Experimental Indicators

In this paper, we evaluate the training test results of the CNN-mLSTM-KAN model using four metrics: accuracy, precision, recall, and F1-Score. Accuracy represents the percentage of traffic—both normal and attack—that is classified as attack traffic out of the total traffic. It is calculated as shown in Equation:

$$A = \frac{T_n + T_p}{T_p + T_n + F_p + F_n} \tag{3}$$

 doi:10.20944/preprints202501.0925.v1

$$R = \frac{T_p}{T_p + F_n} \tag{4}$$

$$F_1 = 2(\frac{PxR}{P + R}) \tag{5}$$

$$P = \frac{T_P}{F_P + T_P} \tag{6}$$

4.3.2. Comparative Experiment

To validate the performance of the AFSB method, we compare it with five representative feature selection techniques. All ranking-based feature selection methods utilize the top ten features for evaluating the proposed model. AFSB consistently demonstrates superior performance when 10, 15, 20, 25, and 30 features are selected for DDoS detection, as shown in Tables 4 and 5. AFSB excels in terms of accuracy, precision, recall, F1 score, loss, and time cost, achieving 99.73% accuracy, 99.47% precision, recall, and F1 score specifically within the DDoS detection model. In contrast, other feature selection techniques also demonstrate significant performance using 10 features. In summary, AFSB can notably enhance detection accuracy while reducing the risk of overfitting compared to traditional feature selection methods and enhancement techniques across multiple standard datasets. This method offers a new perspective and tool for addressing the feature selection problem in high-dimensional data analysis. Therefore, the model is evaluated based on the ten most promising features identified.

Table 4. FS Efficiency in Identifying DDoS Threats.

FS	Number	A	P	R	F1	Loss	ROC Score
ExtraTress	10	99.52%	99.51%	99.49%	99.48%	0.025	99.50%
RandomForest	10	97.24%	97.21%	97.20%	97.19%	0.088	97.08%
XGBoost]	10	99.53%	99.57%	99.58%	99.58%	0.013	99.62%
ANOVA	10	99.43%	99.39%	99.37%	99.37%	0.028	99.41%
LightGBM	10	99.68%	99.56%	99.57%	99.51%	0.014	99.62%
AFSB	**10**	**99.78%**	**99.66%**	**99.65%**	**99.65%**	**0.012**	**99.75%**

Table 5. Impact of Feature Count on AFSB Performance in DDoS Detection.

Performance	Number of feature				
	f=10	f=15	f=20	f=25	f=30
A	99.78%	99.73%	99.74%	99.69%	99.80%
P	99.66%	99.63%	99.64%	99.65%	99.73%
R	99.66%	99.63%	99.64%	99.65%	99.73%
F1	99.66%	99.63%	99.64%	99.65%	99.73%
Loss	0.012	0.011	0.010	0.006	0.008
Time-cost(ms)	0.122	0.124	0.251	0.126	0.224

To validate the performance of the proposed model, this study compares it with other leading machine learning (ML) and deep learning (DL) models applied to DDoS attack detection. The comparison includes two ML-based methods (CART and Extended Decision Tree) and five DL-based methods (LSTM, GRU, CNN, Deep CNN, and the existing CNN-LSTM). To ensure fairness, we utilized the same network configuration for all models and reran the network structures of the existing DL models. We compared their accuracy, computation time, and memory usage, measured in million floating-point operations per second (MFLOPs).

The measurement results show that the accuracies of CNN, LSTM, GRU, and the existing CNN-LSTM models are 99.63%, 99.62%, 99.65%, and 99.78% respectively. Notably, the existing CNN-LSTM

 doi:10.20944/preprints202501.0925.v1

model had a time cost of 0.188 milliseconds and 27.81K learnable parameters. In contrast, the proposed model not only maintained the fewest learnable parameters (only 17.83K) but also reduced the time cost to 0.122 milliseconds while achieving 99.78% accuracy, demonstrating superiority across all metrics.

Moreover, based on the MFLOPs memory measurement findings referenced in [14], Table 6 provides a comparison of the computational complexities between the proposed model and existing models. The MFLOPs for LSTM, CNN [9], Deep CNN [24], the current CNN-LSTM [25], and the proposed model are 5.26, 2.33, 0.829, 0.759, and 0.085, respectively. While models like GRU, LSTM, and CNN [9] achieve strong accuracy in detecting DDoS attacks, they exhibit relatively high MFLOPs and a greater number of trainable parameters. This elevated computational complexity can be attributed to GRU and LSTM employing a large number of hidden layer neurons to process input data directly. In the case of CNN [9], this is due to the cascading of high-dimensional convolutional layers and the extensive use of convolutional kernels (without pooling operations), combined with several fully connected layers, resulting in a significant number of weights and increased MFLOPs. In contrast, the proposed model demonstrates the lowest MFLOPs at just 0.085, signifying minimal memory usage among existing DDoS attack detection models, thus making it more advantageous for implementation in resource-limited software-defined networking (SDN) controllers.

Table 6. Evaluation of Notable ML and DL Algorithms in DDoS Detection.

Techniques	A	Time-Cost	Param	MFLOPS
Extend Decision Tree [34]	97.70%	-	-	-
CART [19]	98.90%	8.40ms	-	-
CNN [10]	99.63%	0.105ms	42.33K	0.829
Deep CNN [25]	99.64%	0.160ms	32.53K	0.759
GRU [10]	99.62%	0.374ms	114.49K	5.26
LSTM [10]	99.25%	0.255ms	51.72K	2.33
Existing CNN-LSTM [26]	99.65%	0.188ms	27.81K	0.521
CNN-mLSTM-KAN	99.78%	0.122ms	17.83K	0.085

Using the same dataset, the model proposed in this paper can better capture the characteristic relationship between flows, so it can be better detected and can achieve better results than other papers.

4.3.3. Ablation Experiment

To demonstrate the effectiveness of each module within the model, ablation studies were conducted on the dataset by removing individual modules while keeping the rest of the structure unchanged. This was to evaluate their impact on the model's performance. The results of the ablation experiments, as shown in Table 7, indicate a decline in performance after the removal of either the KAN or the LSTM modules. This underscores the critical role that both KAN and LSTM play in enhancing the model's detection capabilities. In summary, the ablation studies validate the unique contribution of each module to improving the overall performance of the model.

Table 7. ABLATION STUDY RESULTS for CNN-MLSTM-KAN Model.

Techniques	A	Time-Cost	Parameter	MFLOPS
CNN [10]	99.63%	0.105ms	42.33K	0.829
CNN-mLSTM	99.67%	0.160ms	22.52K	4.58
CNN-KAN	99.64%	0.374ms	21.50K	4.24
CNN-mLSTM-KAN	99.78%	0.122ms	17.83K	0.085

 doi:10.20944/preprints202501.0925.v1

5. Conclusions

With advances in Internet of Things (IoT) technology, the incidence of Distributed Denial of Service (DDoS) attacks has surged, increasingly posing a significant threat to cybersecurity. This research introduces an AFSB feature selection method integrated into a hybrid deep learning detection framework (CNN-mLSTM-KAN), specifically designed for software-defined industrial networks. The architecture of the proposed model comprises three main components: 1) data preprocessing; 2) feature selection; and 3) attack detection. Initially, the AFSB feature selection algorithm is utilized with the CICDDoS2019 dataset to identify the most effective subset of features. Subsequently, the proposed model demonstrates strong performance and reliability across various feature sets.

Experimental results indicate that the CNN-mLSTM-KAN model achieves an impressive accuracy of 99.78%, with a response time of just 0.122 milliseconds. It surpasses existing approaches with only 17.83K learnable parameters and 0.085 MFLOPs. The practicality of the proposed model for real-time IoT applications has been confirmed, underscoring its potential to significantly enhance cybersecurity efforts within industrial environments.

Author Contributions: Conceptualization, X.B. and E.C.; methodology, X.B. and E.C.; software, E.C.and E.C.; validation, E.C. and X.M.; writing—original draft preparation, E.C. ; writing—review and editing, E.C.; funding acquisition, resources, X.B. and E.C; supervision, X.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Research and Development Program of Zhejiang Province (2020C03094), and the General Scientific Research Project of the Department of Education of Zhejiang Province (Y202250677, Y202250706, Y202250679).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. T. Qiu, J. Chi, X.Z.Z.N.M.A.; Wu, D.O. Edge computing in industrial Internet of Things: Architecture, advances and challenges. *IEEE Commun. Surveys Tuts.* **2020**, *22*, 2462–2488.
2. RAHHAL J, S.D. IOT based predictive maintenance using LSTM RNN estimator. *2020 International Conference on Electrical, Communication,and Computer Engineering* **2020**, pp. 1–5.
3. V. Balasubramanian, M.A.; Reisslein, M. An SDN architecture for time sensitive industrial IoT. *Comput. Netw.* **2021**, *186*.
4. Du, M.; Wang, K. An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of Things. *IEEE Trans. Ind. Informat.* **648-657**, *16*, 1.
5. Romero-Gázquez, J.L.; Bueno-Delgado, M. Software architecture solution based on SDN for an industrial IoT scenario. *Wireless Commun. Mobile Comput.* **208**, *2018*.
6. A.A. Pranata, T.S.J.; Kim, D.S. Overhead reduction scheme for SDN-based data center networks. *Comput. Stand. Interfaces* **2019**, *63*, 1–15.
7. W. Mao, Z. Zhao, Z.C.G.M.; Gao, W. Energy-efficient industrial Internet of Things: Overview and open issues. *IEEE Trans.Ind. Informat.* **2021**, *17*, 7225–7237.
8. D. Mourtzis, K.A.; Zogopoulos, V. Mapping vulnerabilities in the industrial Internet of Things landscape. *Procedia CIRP* **2019**, *84*, 265–270.
9. G. C. Amaizu, C. I. Nwakanma, S.B.J.M.L.; Kim, D.S. Composite and efficient DDoS attack detection framework for B5Gnetworks. *Comput. Netw.* **2021**, *188*.
10. N. M. Yungaicela-Naula, C.V.R.; Perez-Diaz, J.A. SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. *IEEE Access* **2021**, *9*, 108495–108512.
11. X. Jing, Z.Y.; Pedrycz, W. Security data collection and data analytics in the Internet: A survey. *IEEE Commun. Surveys Tuts.* **2018**, *21*, 586–618.

12. R. Doriguzzi-Corin, S. Millar, S.S.H.J.M.d.R.a.D.S. LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Trans. Netw. Service Manag.* **2020**, *17*, 876–889.
13. Y. Wei, J. Jang-Jaccard, F.S.A.S.W.X.; Camtepe, S. AE-MLP: A hybrid deep learning approach for DDoS detection and classification. *IEEE Access* **2021**, *9*, 146810–146821.
14. Ziming Liu, Yixuan Wang, S.V. KAN: Kolmogorov–Arnold Networks. *arXiv* **2024**.
15. Maximilian Beck, Korbinian Pöppel, M.S.A.A. xLSTM: Extended Long Short-Term Memory. *arXiv* **2024**, *2405.04517v1*.
16. R. Akter, V.-S. Doan, T.H.T.; Kim, D.S. RFDOA-net: Anefficient ConvNet for RF-based DOA estimation in UAV surveillance systems. *IEEE Trans. Veh. Technol.* **2021**, *70*, 12209–12214.
17. Alzahrani, R.J.; Alzahrani, A. Survey of traffic classification solution in IoT networks. *Int. J. Comput. Appl.* **2021**, *183*, 37–45.
18. L. A. C. Ahakonye, C. I. Nwakanma, J.M.L.; Kim, D.S. Efficient classification of enciphered SCADA network traffic in smart factory using decision tree algorithm. *IEEE Access* **2021**, *9*, 154892–154901.
19. A. O. Sangodoyin, M. O. Akinsolu, P.P.; Grout, V. Detection and classification of DDoS flooding attacks on software-defined networks: A case study for the application of machine learning. *IEEEAccess* **2021**, *9*, 122495–122508.
20. Ullah, I.; Mahmoud, Q.H. Design and development of a deeplearning-based model for anomaly detection in IoT networks. *IEEE Access* **2021**, *9*, 103906–103926.
21. et al., S.H. A deep CNN ensemble framework for efficient DDoS attack detection in software defined network. *IEEE Access* **2020**, *8*, 53972–53983.
22. D. Alghazzawi, O. Bamasaq, H.U.; Asghar, M.Z. Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Appl. Sci.* **2021**, *11*.
23. M. Lopez-Martin, A. Sanchez-Esguevillas, J.I.A.; Carro, B. Network intrusion detection based on extended RBF neural network with offline reinforcement learning. *IEEE Access* **2021**, *9*, 153153–153170.
24. S. K. Sahu, D. P. Mohapatra, J.K.R.K.S.S.Q.V.P.a.N.N.D. A LSTM-FCNN based multi-class intrusion detection using scalable framework. *Comput. Elect. Eng.* **2022**, *99*.
25. A. R. Shaaban, E.A.E.; Hussein, M. DDoS attack detection and classification via convolutional neural network (CNN). *in Proc.9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)* **2019**, p. 233–238.
26. L. Karanam, K.K.P.; Aldmour, R. Intrusion detection mechanism for large scale networks using CNN-LSTM. *in Proc. 13th Int. Conf. Develop. Syst. Eng. (DeSE)* **2020**, p. 323–328.
27. I. Sharafaldin, A. H. Lashkari, S.H.; Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. *in Proc. Int. Carnahan Conf. Security Technol. (ICCST)* **2019**, p. 1–8.
28. A. H. Lashkari, G. Draper-Gil, M.S.I.M.; Ghorbani, A.A. Characterization of Tor traffic using time based features. *in Proc.ICISSP* **2017**, p. 253–262.
29. A. Khraisat, I. Gondal, P.V.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22.
30. M. S. Elsayed, N.A.L.K.; Jurcut, A.D. InSDN: A novel SDN intrusion dataset. *IEEE Access* **2020**, *8*, 165263–165284.
31. Sarica, A.K.; Angin, P. A novel SDN dataset for intrusion detection in IoT networks. *in Proc. 16th Int. Conf. Netw. Serv. Manag. (CNSM)* **2020**, pp. 1–5.
32. A. Aldweesh, A.D.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl. Based Syst.* **2020**, *189*.
33. Y. N. Kunang, S. Nurmaini, D.S.; Suprapto, B.Y. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *J. Inf. Security Appl.* **2021**, *58*.
34. S. Rajagopal, P.P.K.; Hareesha, K.S. Towards effective network intrusion detection: From concept to creation on azure cloud. *IEEE Access* **2021**, *9*, 19723–19742.