An Efficient Hybrid-DNN for DDoS Detection and Classification in Software-Defined IIoT Networks

Ahmad Zainudin, Love Allen Chijioke Ahakonye, *Member, IEEE*, Rubina Akter, Dong-Seong Kim, *Senior Member, IEEE*, and Jae-Min Lee, *Member, IEEE*

***Abstract*—Software-defined networking (SDN)-based Industrial Internet of Things (IIoT) networks have a centralized controller that is a single attractive target for unauthorized users to attack. Cybersecurity in IIoT networks is becoming the most significant challenge, especially from increasingly sophisticated Distributed Denial-of-Service (DDoS) attacks. This situation necessitates efficient approaches to mitigate recent attacks following the incompetence of existing techniques that focus more on DDoS detection. Most existing DDoS detection capabilities are computationally complex and are no longer efficient enough to protect against DDoS attacks. Thus, the need for a low-cost approach for DDoS attack classification. This study presents a competent feature selection method extreme gradient boosting (XGBoost) for determining the most relevant data features with a hybrid convolutional neural network and long short-term memory (CNN-LSTM) for DDoS attack classification. The proposed model evaluated the CICDDoS2019 data set with improved accuracy and low-complexity capability for low latency IIoT requirements. Performance results show that the proposed model achieves a high accuracy of 99.50% with a time cost of 0.179 ms.**

***Index Terms*—Convolutional neural network and long short-term memory (CNN-LSTM), Distributed Denial-of-Service (DDoS) detection and classification, feature selection (FS), Industrial Internet of Things (IIoT), software-defined networking (SDN).**

I. Introduction

THE CONCEPT of Industrial Internet of Things (IIoT) is a large-scale ecosystem that delivers massive sensor and actuator connections, reliable communications, and control systems in industrial applications. These include smart cities, manufacturing, e-healthcare, agriculture, plants, and supply chains. Heterogeneous communication connecting the IIoT architecture enables enterprise networks over global Internet connections. The IIoT system provides machines with intelligence and real-time capability to collect and analyze data for accurate decision making [1]. This reduces human involvement in industrial processes by enabling autonomous operations.

Furthermore, the IIoT enables time-sensitive platforms that require real-time systems and low-latency requirements. The collaboration of the IEEE 802.1 time-sensitive network and open-flow protocols used in a software-defined networking (SDN) environment is the interface to meet the low-latency requirements in IIoT networks [2]. SDN is a promising platform for IIoT networks, providing flexibility, reliability, and efficiency [3]. Several studies have proposed SDN-based network architecture solutions for IIoT scenarios by considering the advantages of SDN. This adoption has been increasingly impacted, especially for medium-high factories, by the heterogeneity of connected IoT devices [4]. SDN allows a centralized controller to manage network devices and separate the data and control planes. IIoT facilities can be effectively managed and configured rapidly [5]. Moreover, the SDN virtualization component can reconfigure network devices, traffic, and bandwidth provision.

Network security is a major concern in the development of IIoT networks. IIoT systems allow the transmission of vast amounts of sensitive data to the cloud, which increases their vulnerability to attacks [6]. In addition, SDN-based IIoT networks have a centralized controller, which is an attractive target for unauthorized users to attack. Cybersecurity in IIoT networks is the most significant challenge, especially security from Distributed Denial-of-Service (DDoS) attacks at the network and application layers [7]. This attack can disrupt the target server by sending many request packets and congest the network by flooding malicious network packets with forged source addresses [8]. DDoS attacks occur in the network, transportation, and application layers. The network layer DDoS threats include Internet control message protocol flooding, whereas the transport layer flooding threats are the synchronize (SYN) and user datagram protocol (UDP). Application-layer DDoS flooding attacks are more sophisticated and should be handled with a special treatment [9]. DDoS attacks usually utilize the weaknesses of the network/transport protocol to attack target servers. Therefore, it is crucial to facilitate IIoT networks with an intrusion

Manuscript received 30 March 2022; revised 11 July 2022; accepted 25 July 2022. Date of publication 8 August 2022; date of current version 9 May 2023. This work was supported in part by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT), South Korea, under Grant 2018R1A6A1A03024003, and in part by the Grand Information Technology Research Center Support Program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2022-2020-0-01612. *(Corresponding author: Jae-Min Lee.)*

Ahmad Zainudin is with the Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea, and also with the Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya 60111, Indonesia (e-mail: zai@kumoh.ac.kr).

Love Allen Chijioke Ahakonye, Dong-Seong Kim, and Jae-Min Lee are with the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea (e-mail: loveahakonye@kumoh.ac.kr; dskim@kumoh.ac.kr; ljmpaul@kumoh.ac.kr).

Rubina Akter is with the Department of IT Convergence Engineering and the ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, Gyeongsangbuk, South Korea (e-mail: rubinaakter2836@kumoh.ac.kr).

Digital Object Identifier 10.1109/JIOT.2022.3196942

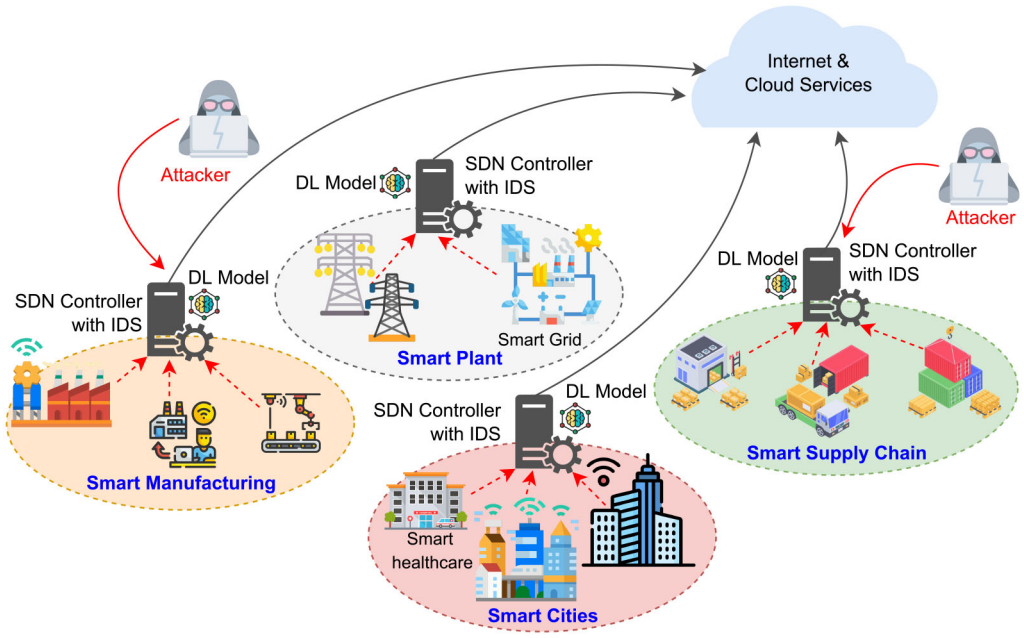

Fig. 1. SDN-based IIoT network with attacks.

detection system (IDS) that allows the privacy, integrity, and security of IIoT networks.

To overcome these threats, the work [10] discussed some approaches to DDoS flooding attacks detection and classification in IoT networks. DDoS flooding attacks detection using statistical, machine learning (ML), and knowledge-based techniques have been described. However, these statistical approaches may not apply to online systems. The features used for detection parameters, such as the number of transmitted packets, total bytes, and flow duration between the source and destination, cannot be processed in real time by the system [11]. ML-based DDoS detection provides high detection accuracy. However, the detection performance degrades as the feature dimensions increase. Furthermore, this approach requires a separate feature-extraction process when using unlabeled data sets. These limitations were addressed by leveraging a deep learning (DL) method to detect and classify DDoS attacks [12] networks.

These problems persist, even though the DL approach is used. DL-based solutions require significant resources to compute and learn the detection and classification tasks [9], [13]. To manage networks effectively, the SDN controller handles all communication between devices and applications. To accomplish this task, a controller executes some services on the application plane, such as network monitoring, flow management, load balancing services, IDSs, etc. The controller in the SDN-based IIoT network is illustrated in Fig. 1. It captures network traffic data, detects, and classifies malicious traffic in each edge network server. It then decides on its defenses or forwards the classified result to the cloud server. A high-computational model can overload the SDN controller. Therefore, lightweight DDoS attack detection and classification is promising for SDN-based IDS systems in IIoT networks.

Considering the requirements for advanced DDoS classification with low latency capability in an industrial IoT network environment, this study proposes the following contributions.

1) We used various ML approaches, such as extremely randomized trees (Extra Trees), random forest (RF), extreme gradient boosting (XGBoost), mutual information (MI), analysis of variance (ANOVA), and the light gradient boosting algorithm (LightGBM) to explore the most efficient feature selection (FS) strategy.
2) We analyzed the FS techniques in terms of the accuracy, precision, recall, F1-score, loss, and time cost. Based on these performance metrics, we chose the XGBoost algorithm to reduce the feature dimensions. This FS strategy offers the ability to select the prominent features and assists in the achievement of robust performance.
3) We proposed a hybrid deep neural network (DNN), deploying both the convolution neural network (CNN) and the long short-term memory (LSTM) layer. The proposed model efficiently learns the intrinsic DDoS attack features in a cost-intensive manner. The CNN is designed with a low computing convolution layer using the concept of factorization. Moreover, residual connectivity is adopted to enhance the learning efficiency and resolve the vanishing gradient problem. Notably, the proposed model exploits the LSTM layer to accurately detect DDoS attacks.
4) In simulation, we investigated the robustness of the proposed model by analyzing two tiers of classification, such as DDoS detection and type identification. The analysis was conducted through a series of experiments with the benchmark CICDDoS2019 data set. Finally, the performance is evaluated by comparing our result with that of state-of-the-art solutions.

The remainder of this article is structured as follows: Section II presents the state-of-the-art of the proposed model for DDoS detection and classification is described in Section III. The experiment results and discussion are presented in Section IV. Finally, Section V concludes the study and discusses future research directions.

II. State-of-the-Art DDoS Detection and Classification

Generally, DDoS detection and classification follow three techniques: 1) statistical; 2) ML; and 3) DL approaches [14]. Some of these approaches have crucial weaknesses and are not suitable for real-time DDoS detection and classification scenarios in IIoT with low latency requirements. Feature engineering lacks major considerations, which is fundamental to the processing of data generated from heterogeneous applications. This section presents the sneak peeks of the different approaches mentioned.

A. Statistical Methods

Statistical analysis methods often exploit connection-level data, packet-level data, and flow-rate data for DDoS attack detection [10]. Nijim *et al.* [15] implemented a data-mining engine platform to capture, analyze, and predict network traffic, whether malicious or legitimate requests, based on data correlation. Packet-level attack detection can detect malicious traffic when traffic volume exceeds a determined threshold. In addition to the achievements of this approach, it is limited by the inability to withstand overwhelming attack floods, which are likely to break a predetermined threshold. Maity *et al.* [16] proposed a probabilistic-based DDoS detection method in SDN environments. This approach used the central limit theorem to analyze the outcome and probability theory to identify DDoS attacks. These techniques are primarily intended to identify resource depletion attacks. The proposed method exploited the flag distribution in the transmission control protocol (TCP) header of incoming packets to train in offline mode.

B. ML Approaches

ML techniques have been promising for detection and classification problems [17]. Ahakonye *et al.* [18] explored the efficiency of ML classifiers for the classification of enciphered supervisory control and data acquisition (SCADA) network traffic in smart factories. In [13], the study investigated a low-cost ML-based scheme to detect and classify the DDoS flooding attacks on an SDN architecture. TCP, UDP, and HTTP flooding attacks represent the highest traffic usage of a Web application applied in the proposed IDS. Some ML-based approaches for DDoS classification that have been utilized include the use of the Gaussian naive Bayes (GNB), classification and regression tree (CART), k-nearest neighbor (k-NN), and quadratic discriminant analysis. These approaches have quite enabled the achievement of the best performance with an accuracy of 98% for the CART technique. However, these approaches require a computation time of up to 12.4 ms, which is high for a real-time operation of the IIoT applications [13].

C. DL Approaches

To resolve attacks on DDoS, various studies have presented DL approaches that are widely accepted for their robustness and ability to learn and predict meaningful attributes from the network traffic [19], [20], [21]. Wei *et al.* [12] implemented a hybrid DL autoencoder multilayer perceptron network (MLP) with automatic feature extraction capabilities. The autoencoder extracts important features through compressed and feature-reduced scenarios. The system used the CICDDoS2019 data set and achieved an accuracy rate of 98.34%. Other studies evaluated the same data set [9], [22]. Amaizu *et al.* [8] applied a composite DL model to classify DDoS attacks on B5G networks. The proposed model was developed by concatenating two DNNs with different structures. This is because the DNN used in this system is sufficiently deep to seven layers for each framework, resulting in a high computation time.

An IDS with experimental approaches has been widely proposed. Yungaicela-Naula *et al.* [9] implemented a modular SDN-based network to address application- and transport-layer DDoS attacks by applying the ML/DL model. The proposed model was tested using the latest CICDDoS2017 and CICDDoS2019 security data sets. The system performance was validated through experimental measurements in a simulated SDN environment using Mininet simulator and open network operating system (ONOS) as SDN controllers. This experimental setup is closest to the proposed system. Transport-layer DDoS attacks, such as SYN and UDP floods, were evaluated and considered application-layer DDoS attacks. The system investigated a support vector machine (SVM), RF, KNN, CNN, MLP, LSTM, and gated recurrence unit (GRU). This model achieved a detection rate of 95% for slow-rate attacks and 98% for high-volume attacks. Sahu *et al.* [23] implemented a hybrid LSTM and fully connected network (FCN) with hyper-parameters tuning to classify benign and malicious network traffic activities. This approach considered the imbalanced intrusion data distributions for the majority and minority classes. This approach evaluated binary and multiclass intrusion detection scenarios by utilizing and comparing six cyber security data sets.

Limitations, such as high computational complexity and lack of an efficient model for advanced DDoS attacks are prevalent in existing studies on ML/DL approaches [9], [24], [25]. Therefore, this study proposes an efficient IDS technique for DDoS detection and classification with low latency and low complexity in IIoT networks. Fig. 2. shows an overview of the proposed system. A network monitoring system running in an SDN controller recursively captures IIoT network traffic, including benign and malicious traffic. The attacker executes several DDoS attacks, including reflection- and exploit-based attacks. The network equipment (router or switch) captures all incoming and outgoing traffic as shown in Fig. 3. The captured data set was extracted using CICFlowMeter to generate the traffic features [26]. A potential FS mechanism was applied to obtain important features and enhance the accuracy of the proposed model.

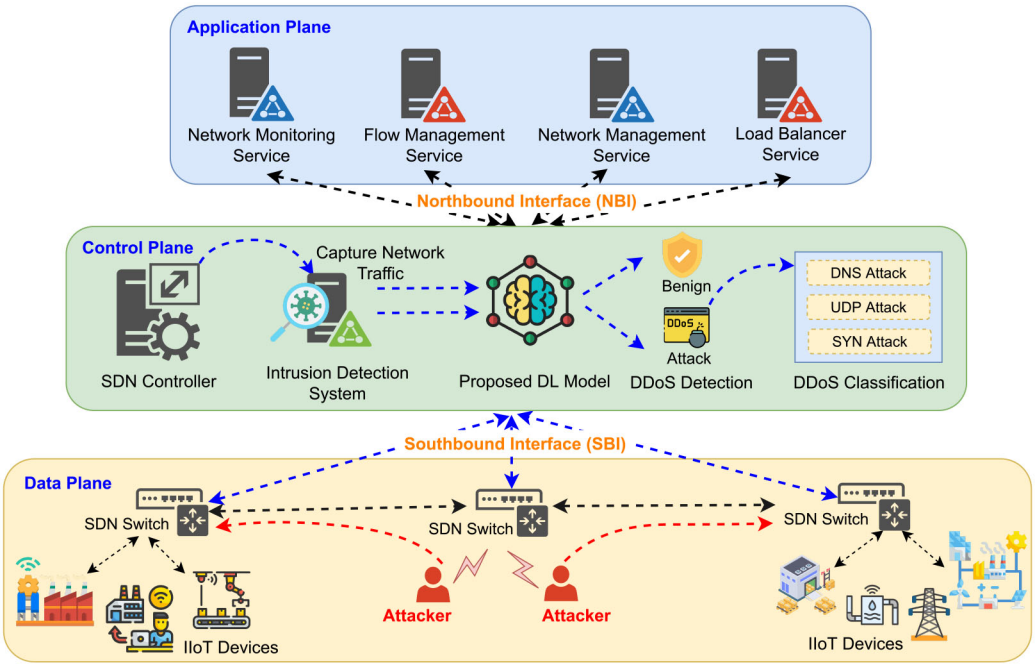

Fig. 2. Proposed DDoS detection and classification in SDN-based IIoT networks.

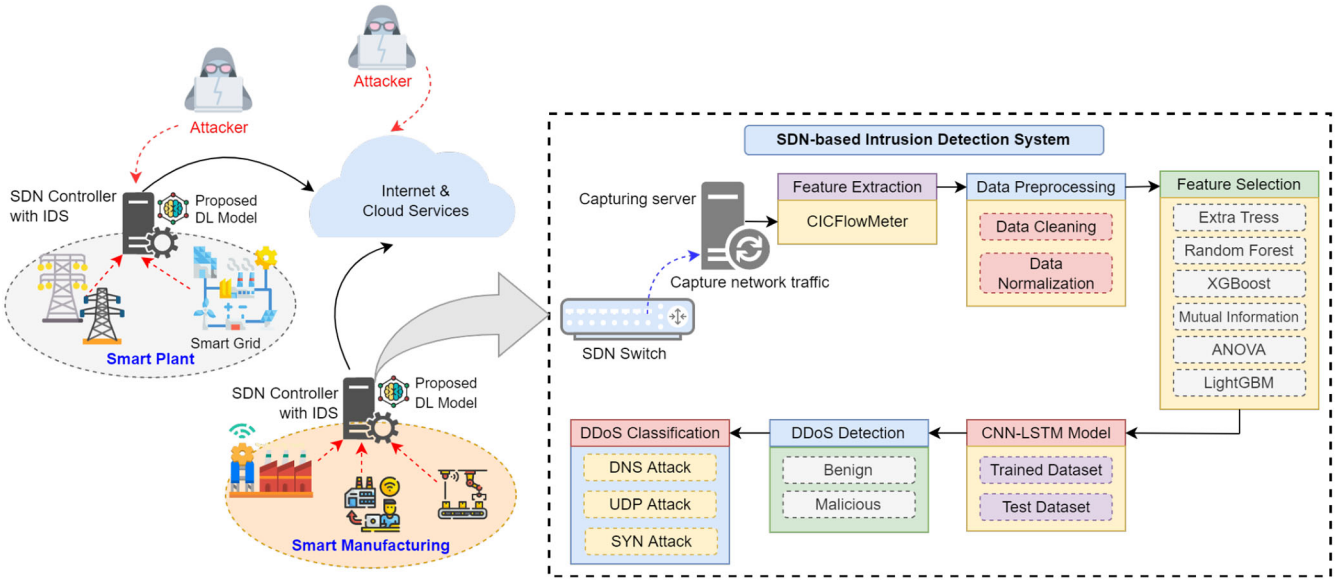

Fig. 3. Workflow diagram for DDoS detection and classification using CNN-LSTM.

III. Proposed Approach

As the center of network management, The SDN controller must handle all communication between applications and network devices. Fig. 2 shows the proposed SDN-based DDoS detection and classification in IIoT networks. Northbound interfaces allow the controller to communicate with applications, such as network monitoring, flow management, network management, load balancer services, and network IDS. A southbound interface, such as the OpenFlow protocol, permits the controller to communicate with particular network devices in the data plane. Using these southbound protocols, the controller can organize network devices and select the optimum network connectivity route for application traffic. A reliable and low-complexity IDS with classification capacity is required due to the vast number of tasks that the SDN controller must manage. In this article, we propose a convolutional neural network and LSTM (CNN-LSTM)-based DDoS detection and classification method in an IIoT network environment.

The workflow of the work diagram used in this study is shown in Fig. 3. In the first step, a capturing server exploits CICFlowMeter [27] to capture network traffic on the core switch and extract the original 88 traffic features using a pcap file. Preprocessing is required to ensure great data quality before supplying the suggested model with the data, including data cleaning and normalization. The proposed model was

applied to the SDN controller. This model leverages several FS techniques to extract the most distinct features that significantly contribute to achieving more excellent performance in DDoS attack detection. The feature-selection mechanism exploited the ML classifier, such as Extra Trees, RF, XGBoost, MI, ANOVA, and LightGBM, empowered to select the most promising features. The selected features are processed into the proposed model to classify the traffic as benign or malicious, such as domain name system (DNS), UDP, or SYN attacks.

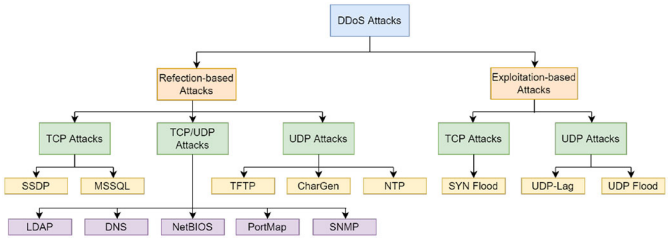

Fig. 4. DDoS attacks categorization and taxonomy [22].

A. Problem Statement

The established algorithms enable intrusion detection for common attacks (i.e., DoS, DDoS, botnets, brute force, etc.) in the network security domain, assessed using outdated data sets, such as NSL-KDD or KDD99, publicly released from 1999 to 2009 [28], [29], [30]. The most recent DDoS attacks in the current infrastructure environment are not represented in these data sets. These studies missed DDoS attacks that follow a specific attack pattern. DDoS attacks have evolved to become not only more frequent and dangerous but also more intelligent. Presently, modern and more sophisticated DDoS attacks, such as UDP, DNS, SYN, and network time protocol (NTP) [26] threaten server resources in IIoT networks. Existing approaches focus on DDoS detection. DDoS detection capabilities are not sufficiently effective to protect against DDoS attacks. A DDoS attack classification capability using a data set of recent DDoS attacks is required to determine the appropriate defense system. Different DDoS attacks require different defense responses [31].

The performance of existing DDoS classifications degrades as the number of feature dimensions increases. The existing DDoS classification model with complex architecture require more computation [9], [24], [25], which can be considerably reduced by optimizing FS to slash the number of network traffic features [32]. Potential FS mechanisms utilizing DL are used to classify DDoS attacks with high-dimensional feature data sets. A great FS mechanism can improve detection performance and reduce model complexity. In addition, a lightweight DDoS detection and classification model architecture is required for security systems in IIoT networks with low-latency requirements.

B. Data Set Description and Preprocessing

For IDS evaluation, there are some publicly available data sets (i.e., DARPA/KDD99, CAIDA, NSL-KDD, ISCX 2012, ADFA-LD, and ADFA-WD) [33]. IDS data sets are recommended for the SDN architecture scenario [34], [35] that is listed in [28] and [29]. However, all of these data sets are unable to accommodate the recent DDoS attack categorization. Some researchers [9], [36], [37] continue to assess their SDN-based DDoS classification approaches using the latest CICDDoS2019 DDoS data set. This data set contains 88 SDN network-related traffic features that are easy and simple to process and extract sophisticated features using SDN controllers [38].

This study utilized the CICDDoS2019 [26] data set which contains the most recent DDoS attacks and a benign sample from the Canadian Institute for Cybersecurity. The network traffic CICDDoS2019 data set contains a large number of network traffic samples and features compared to the existing data set. The data set contains raw network traffic data in PCAP format and other format flow-based features extracted using CICFlowMeter in a CSV file. The flow-based data set contains 88 characteristics and 12 different DDoS attacks, such as UDP, UDP-Lag, SYN, lightweight directory access protocol (LDAP), DNS, microsoft structured query language (MSSQL), trivial file transfer protocol (TFTP), network basic input/output System (NetBIOS), simple service discovery protocol (SSDP), simple network management protocol (SNMP), NTP, and WebDDoS.

The CICDDoS2019 data set considers some reflection- and exploitation-based DDoS attacks [26]. Fig. 4 shows the full taxonomy of DDoS attacks.

1) *Reflection-Based DDoS Attacks:* In these attacks, legitimate third-party elements such as reflector servers are used to camouflage the identity of the attacker. Attackers send the packets to reflector servers with the source IP address set to the IP address of the target victim to flood the victim with response packets. Reflection-based DDoS attacks perform by utilizing application and transport layer protocols, such as UDP, TCP, or a mix of the two. TFTP, NTP, and CharGen are just a few examples of UDP-based DDoS attacks, whereas TCP-based protocols include SSDP and MSSQL attacks. A combination of TCP and UDP can perform attacks on DNS, LDAP, NetBIOS, and SNMP. In a DNS attack, attackers use a flooding attack on specific DNS servers to obstruct DNS resolution for a given domain. A DNS flood attack will damage a website application's capacity to respond to legitimate traffic since it will disrupt DNS resolution.
2) *Exploitation-Based DDoS Attacks:* Exploitation-based attacks use the application, transport, and network layer protocols, such as TCP and UDP to exhaust the victim's resources by overwhelming the network bandwidth with massive network packets [39]. TCP SYN flood is a TCP-based exploitation attack, whereas UDP-based exploits comprise UDP flood and UDPLag. Attackers disrupt the victim's legitimate connection by leveraging TCP SYN flooding from the weakness of the procedure to establish a three-way handshake connection. In a TCP SYN attack, the attackers initiate by delivering a huge number of SYN packets to a server and then

TABLE I
DISTRIBUTION OF THE SAMPLES CICDDOS2019 DATA SET

Class	Sample
Benign	36082
DNS Attack	33900
UDP Attack	35640
SYN Attack	33217
Total	**138839**

refusing to respond to the server's reply. Because the server's restricted buffer queue is full, no additional connection requests may be processed in this state. A UDP flood attack is launched by sending a huge number of UDP packets to the remote host. These UDP packets are transmitted speedily to random ports on the target system. Consequence, the network's available bandwidth is depleted, the system crashes, and performance decreases.

The proposed model was evaluated by leveraging three DDoS attack types (DNS, UDP, and SYN) and benign network traffic. The data set used had 138 839 samples containing benign traffic and some attack activity (i.e., DNS, UDP, and SYN attacks) as shown in Table I. The DDoS attacks used in this study are described as follows.

1) *DNS Attack:* DNS attack leverages the functionality of open DNS resolvers to overwhelm the target server.
2) *UDP Attack:* A UDP flooding attack, is initiated by transmitting a huge number of UDP packets to a disordered port of the target server. This attack can exhaust the network bandwidth, thereby causing the target system to crash.
3) *SYN Attack:* SYN flooding attack uses the TCP-three-way handshake protocol and bombards the target system with SYN packet requests.
4) *Benign Network:* Normal network traffic human activities which do not contain a DDoS attack.

Preprocessing was required to achieve good data quality before feeding into the proposed model. In this step, several processes were performed, such as data cleaning and normalization.

1) Data Cleaning: The initial flow-based data set comprised 88 features. Some noncontributing features were removed. Features, such as "Unnamed," "Timestamp," "Flow ID," "Destination Port," "Source Port," "Destination IP," "Source IP," and "SimilarHTTP." After removing the eight features, we obtained 80 features for further work. To cleaning up NaN-contained values, infinity, and empty values are also required. This data set contains large samples; therefore, we can remove samples with malformed values.

2) Data Normalization: Several features (such as "Flow Duration," "Flow IAT Max," "Flow IAT Std," and "Bwd IAT min") have a high variance between the minimum and maximum values. Data normalization was performed to avoid high variance values across the features. In addition, this process can reduce the training processing time and improve the model performance. In this study, MinMax-based normalization was implemented for feature scaling. This process is based on

$$X_{sc} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

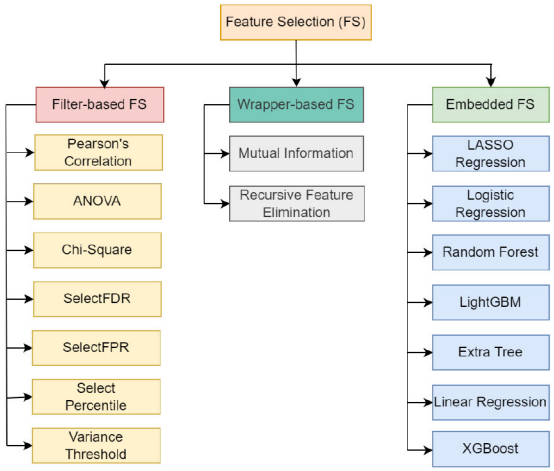

Fig. 5. FS categorization [40].

where X_{sc} represents the normalized numeric results ranging from [0,1]. max and min refer to the maximum and minimum values of the represented features, respectively.

C. Feature Selection

DDoS detection and classification models degrade as feature dimensions increase. Some features had little or no correlation with the DDoS classification process. Reducing the feature dimensions to eliminate redundant data sets is necessary to improve model capability and reduce model training time. FS methods can be categorized into three: 1) wrapper based; 2) filter based; and 3) embedded based. The wrapper-based FS technique works based on a particular classification method to recognize a feature and update the feature subset based on the learning model. The filter-based FS technique selects potential features that are highly correlated with the target parameter. This method allows for faster execution with low computational complexity. The embedded FS technique exploits filter- and wrapper-based methods to determine potential features [40]. In this study, six FS algorithms were used (Fig. 5, Tables VI and VII).

1) Extra Trees FS Technique: This technique is patterned after the ensemble learning approach, which collects the outcomes of numerous de-correlated decision trees aggregated to produce the classification outcome. This approach is comparable to the RF technique, except for the decision-tree formation method [40]. The Extra Trees FS technique selects a distinct feature set that appropriately qualifies the target classification.

2) Random Forest FS Technique: An FS approach employing RF is offered by a class of embedded techniques that unify the attributes of wrapper and filter methods. RF is a prevalent ML algorithm with competent predictive achievement, ease of interpretation, and low overfitting. Its interpretability is inclined toward the fact that it is difficult to derive the importance of each variable [41]. This method is based on a decision tree; however, following the presence of n trees, it constitutes a forest. This approach combines the prediction outcomes from various trees and takes the most selected as the prediction outcome; hence, the name RF [42].

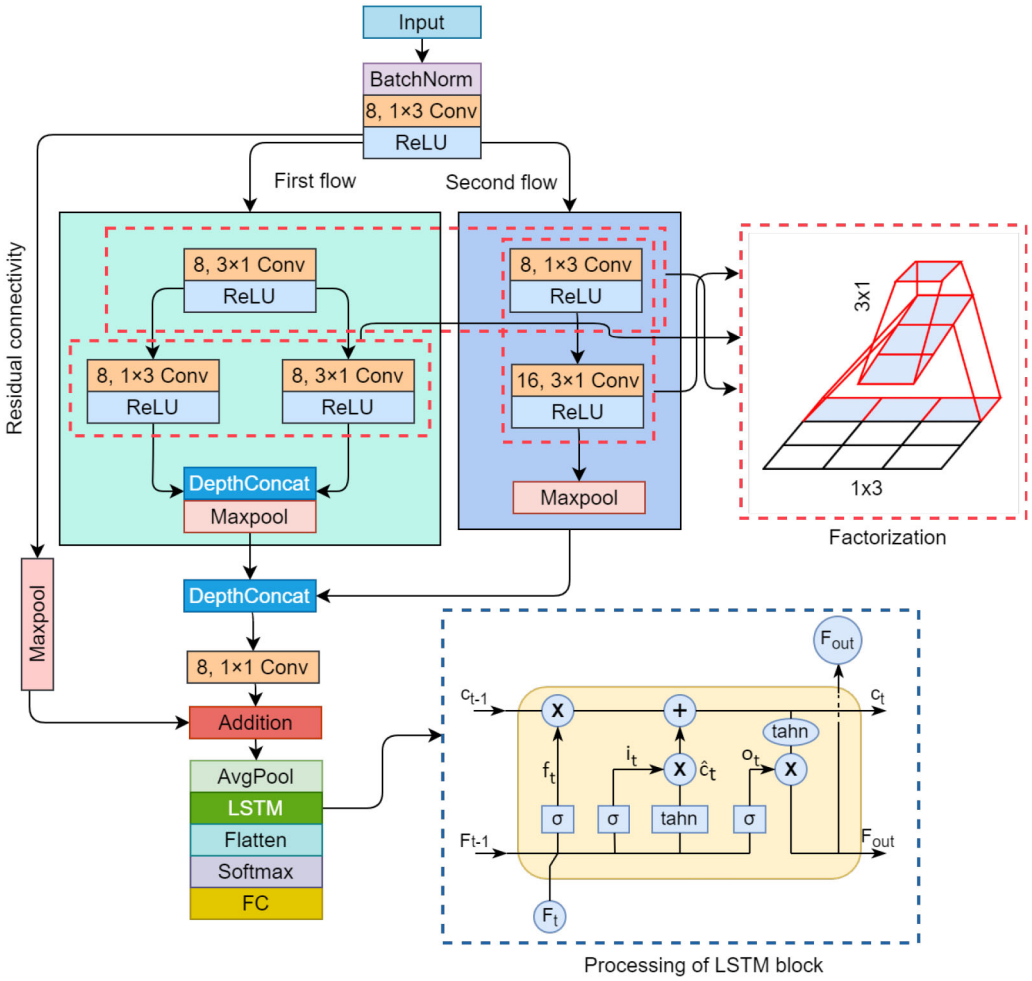

Fig. 6. CNN-LSTM-based proposed model.

3) XGBoost FS Technique: XGBoost is a decision-tree-based ensemble ML technique built on a more improved gradient boosting structure [43]. The benefits of using XGBoost include the speed of execution, high scalability, and better performance compared with other algorithms, such as gradient boosting, decision tree, and RF, and the consistent normalization of the model in controlling overfitting.

4) Mutual Information FS Technique: The MI FS technique has been successfully adopted to assess the redundancy and relevancy of feature subsets with other variables. MI is measured between two variables, measuring the minimization of uncertainty for one variable given the value of the other variable. Accordingly, for variables X; Y, the MI between them can be symmetric, and is denoted by $I(X; Y) = I(Y; X)$ [44]. MI is computed using two variables, and the uncertainty reduction for one variable is evaluated, given the known value of the other variable.

5) ANOVA FS Technique: ANOVA is a statistical approach used to determine the most promising features, specifically used in FS. It constitutes clusters conforming to the categorical label and considers the deviation between the balance. A higher variation indicates better server ability, which implies that they are more suitable for detection [45].

6) LightGBM FS Technique: LightGBM is part of a decision tree for ranking, regression, and classification tasks. This classifier calculates the variance gain between the small and large gradients. LightGBM overcomes big data and has high feature dimensions.

D. Proposed CNN-LSTM-Based Model Architecture

This section presents the proposed hybrid DNN, which is configured with both CNN and LSTM layers; it challenges the accuracy and computational complexity of existing DDoS classification techniques in SDN-based IIoT networks. Based on numerous studies, an increase in network size and computational expense translates to immediately achieving good classification accuracy; however, computational cost and low parameter count are still enabling issues in DDoS attack detection in IIoT platforms. To mitigate the aforementioned issue, this study explores factorized convolution and skip connectivity to utilize proper parameter aggregation and achieve learning efficiency at a reduced computational expense. As shown in Fig. 6, the proposed model is constituted using a regular convolution layer incorporated to generate the disentangled features and an LSTM layer for traffic classification (benign, DNS attack, UDP attack, or SYN attack).

First, input data were preprocessed using the MinMax data normalization technique and the input layer was configured as $\mathbf{I} \in \mathrm{N}^{\mathrm{batchsize}\times 10\times 1}$. A batch normalization layer was deployed to accelerate the convergence of input data, which assists the model in preventing overfitting. In this study, the input data were standardized into layers for each of the 32 mini-batch sizes. This standardization technique followed a two-step process: 1) normalization and 2) rescaling and offsetting. To perform normalization, the mean of the hidden activation μ is calculated as follows:

$$\mu = \frac{1}{n}\sum l_i \tag{2}$$

where n is the number of activation in the layer l_i. Consequently, the batch normalization layer calculates the standard deviation σ using the mean function as follows:

$$\sigma = \sqrt{\frac{1}{n}\sum (l_i - \mu)^2}. \tag{3}$$

Finally, the normalization of hidden activation $l_i(norm)$ can be expressed as follows:

$$l_{i(\mathrm{norm})} = \frac{l_i - \mu}{\sigma + \varepsilon} \tag{4}$$

where ε is the smoothing term that ensures numerical stability within the operation. In the final stage, the rescaling and offsetting of the input data occurs through the rescaling and shifting parameters as follows:

$$l_i = \gamma l_{i(\mathrm{norm})} + \beta. \tag{5}$$

Thus, the output of the batch normalization layer is processed through the (1×3) convolution layer employing eight kernels. In the convolutional layer, the 1-D convolution operation of the kernel and input map is the sum of the dot products at a specific spatial coordinate (x,y), as follows:

$$\mathrm{conv}_{x,y} = \sum_{l_i} w_{l_i} v_{l_i} \tag{6}$$

where w_{l_i} refers to the convolution kernel weight and v_{l_i} is the network traffic value of the input data. Scalar bias value b determines the convolution output and can be added as a determined value

$$z_{x,y} = \mathrm{conv}_{x,y} + b. \tag{7}$$

The output dimension of z is $\mathrm{N}^{\mathrm{batchsize}\times 10\times 8}$, where 8 is the number of channels used.

The feature map is processed using a nonlinear activation function g as the subsequent convolution layer. A rectified linear unit (ReLU) activation function was applied to reduce the over-fitting problem [46]. The ReLU generates a zero value if any value is less than zero and passes the value equal to ore greater than zero. The operation of the ReLU is as follows:

$$g(z) = \begin{cases} z, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0. \end{cases} \tag{8}$$

The output feature map of the ReLU layer was processed using three flow connectivity mechanisms. The processing unit of the first and second flows are combined by the concatenation layer, where these output feature maps are combined with the previous block output via the residual connectivity. In summary, the output feature map of the ReLU layer is received by the (3×1) (first flow) and (1×3) (second flow) convolution layers. Both convolution layers are configured with a stride of (1×1) and eight kernel numbers. The aim of using (1×3) kernel followed by a (3×1) convolution kernel is equivalent to skating a two-layer network with the same kernel as (3×3) convolution, as shown in Fig. 6. Furthermore, this concept, known as factorization, offers greater learning efficiency and reduced costs. In the first flow, the output feature map of the convolution layer is again processed with an asymmetric convolution kernel, which assists in extracting depth features and enhancing model accuracy. Subsequently, a depth-wise concatenation layer concatenates the output volume of the asymmetric convolution layers. The operation of the concatenation layer can be expressed as follows:

$$F_{\mathrm{concat}} = \mathcal{D}\left(X^{f2}_{1\times 3}\left(X^{f1}_{3\times 1}\right), X^{f2}_{3\times 1}\left(X^{f1}_{3\times 1}\right)\right) \tag{9}$$

where $X^{f2}_{1\times 3}$ and $X^{f2}_{3\times 1}$ represent the second factorization block outputs of 1×3 and 3×1 convolution layers, respectively, which obtain the input from the first factorized block input $X^{f1}_{3\times 1}$. $\mathcal{D}$ represents the depthwise concatenation function. Subsequently, a maxpool layer processes the output from the concatenation layer. Notably, the maxpool layer is arranged as a pool size (2×1) which reduces half of the input parameters. Therefore, the output size of the maxpool layer is $\mathbf{I} \in \mathrm{N}^{\mathrm{batchsize}\times 5\times 1}$, which is processed by the subsequent layers.

In terms of the second data processing flow, a 1×3 convolution layer was deployed to process the output of the ReLU layer. subsequently, we employed a 3×1 convolution layer with a 16 kernels to extract more depth features. These two asymmetric convolutional kernel extract the features in parallel by traversing the receptive field along the vertical and horizontal axes. A depthwise concatenation layer is organized to obtain more diverse features generated from the connections between the two flows. Notably, to scale the input of the concatenation layer, we employed a maxpool layer with a pool size of 2×1 that reduces the input dimension and contributes to reducing the computation complexity of the successive layers as follows:

$$\mathcal{F}_{\mathrm{pool2}} = \mathcal{P}\left(X^{f3}_{3\times 1}\left(X^{f1}_{1\times 3}\right)\right) \tag{10}$$

here, $\mathcal{F}_{\mathrm{pool2}}$ is the output of the maxpool layer, which aggregates the output features of the asymmetric convolutions presented as $X^{f3}_{3\times 1}(X^{f1}_{1\times 3})$. The operation of the concatenation layer F_{concat2} is expressed as follows:

$$F_{\mathrm{concat2}} = \mathcal{D}\left(\mathcal{F}_{\mathrm{pool2}}, \mathcal{P}(F_{\mathrm{concat}})\right) \tag{11}$$

subsequently, a unit convolution layer expands the features collected from the concatenation layer. Furthermore, a skip connection leverages the maxpool layer with the output features of the unit convolution layer. This connectivity offers faster convergence and resolves the vanishing gradient problems in the proposed network. Skip connectivity F_t can be expressed as follows:

$$F_t = \mathcal{P}(z) + X_{1\times 1}(F_{\mathrm{concat2}}) \tag{12}$$

where $\mathcal{P}(z)$ is the residual connection and $X_{1\times1}(F_{\text{concat2}})$ are the features of the unit convolution layer. The aggregated features of the addition layer are fed to an average pool layer that calculates the average value and downsamples the features.

Owing to the efficient mechanism of the LSTM layer in capturing long-term temporal dependencies, the proposed model adopts an LSTM layer after the average pool layer. The LSTM layer contains neurons known as memory cells [47]. Each memory cell has three gates: 1) input; 2) forget; and 3) output gates. Each gate has different functionalities to process the input features. For instance, the forget gate selects information that should be processed by removing unnecessary information depending on the state of the cell. First, the forget gate processes the addition layer output F_t and the previous cell output F_{t-1} through a sigmoid gate σ to eliminate unnecessary data. Finally, the obtained data are merged by multiplication. The output of the forget gate f_t is expressed as follows:

$$f_t = \sigma\Big(W_f.\big[F_{t-1}, F_t\big]^T + b_f\Big) \tag{13}$$

where W_f and b_f are the weight matrix and the offset of the forget gate, respectively.

Consequently, the input gate employs a sigmoid function to control the input data and generate the current state. The processing of the input gate i_t can be expressed as follows:

$$i_t = \sigma\Big(W_i.\big[F_{t-1}, F_t\big]^T + b_i\Big) \tag{14}$$

where W_i, and b_i are the weight matrix and the offset of the input gate, respectively. Moreover, the input gate employs a tanh function to generate a vector of the information to be added to the current state. Using both the outputs of the forget and input gates, the proposed network determines the hidden state $\tilde{c_t}$ as follows:

$$\tilde{c_t} = \tanh\Big(W_c.\big[F_{t-1}, F_t\big]^T + b_c\Big) \tag{15}$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c_t} \tag{16}$$

Finally, the output gate selects useful features based on the current state of the cell, the result obtained from the previous cell, and new data. The output function of this gate o_t can be expressed as follows:

$$o_t = \sigma\Big(W_o.\big[F_{t-1}, F_t\big]^T + b_o\Big). \tag{17}$$

The output of the LSTM layer F_{out} can be expressed as follows:

$$F_{\text{out}} = o_t \times \tanh(c_t). \tag{18}$$

The output LSTM layer is fed into the dense layer. After the dense layer, the output data are processed by softmax and fully connected layers to classify each attack category. The detailed network structure of the proposed model is shown in Table II.

The optimal parameters setting of the proposed model is described in Table III. The proposed model performs accurately by following the training configurations: ten selected features, a mini-batch size of 32 for 50 epochs, an initial training rate of 0.001, ReLU is used as the activation function, adam optimizer with cross-entropy loss function, and the

TABLE II
NETWORK STRUCTURE OF THE PROPOSED MODEL

Component	Output Dimesion	Detailed Description
Input	[batchsize, 10, 1]	
Batch Normalization	[batchsize, 10, 1]	
Conv	[batchsize, 10, 8]	8, 1x3, ReLU, padding
Conv	[batchsize, 10, 8]	8, 1x3, ReLU, padding
Conv	[batchsize, 10, 16]	16, 3x1, ReLU, padding
MaxPool	[batchsize, 5, 16]	Pooling size=2
Conv	[batchsize, 10, 8]	8, 3x1, ReLU, padding
Conv	[batchsize, 10, 8]	8, 1x3, ReLU, padding
Conv	[batchsize, 10, 8]	8, 3x1, ReLU, padding
DepthConcat	[batchsize, 10, 16]	
MaxPool	[batchsize, 5, 16]	Pooling size=2
DepthConcat	[batchsize, 5, 32]	
Conv	[batchsize, 5, 8]	8, 1x1, ReLU, padding
MaxPool	[batchsize, 5, 8]	Pooling size=2
Addition	[batchsize, 5, 8]	
AveragePooling	[batchsize, 2, 8]	Pooling size=2
LSTM	[batchsize, 2, 25]	25 neuron, ReLU
Flatten	[batchsize, 50]	
Fully-connected	[batchsize, 4]	4 nodes, softmax

TABLE III
OPTIMAL PARAMETER SETTING OF THE PROPOSED MODEL

Parameters	Value
Total Features	80
Selected Features	10
Optimizer	Adam
Learning rate	0.001
Loss Function	Cross-entropy loss
Epoch	50
Batch Size	32
Activation Function	ReLU
Cross Validation	5 k-fold

k-fold for cross-validation is 5. The training was conducted using Google Colaboratory in a Keras environment with 8-GB VRAM and NVIDIA GeForce GTX 1050.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This section reports the experimental setup and simulation results to comprehensively evaluate the robustness of the proposed model. The system model is evaluated using performance classification metrics such as accuracy A, precision P, recall R, F1-score F_1, and time cost using the following formulas:

$$A = \frac{T_p + T_n}{T_p + F_n + F_p + T_n} \tag{19}$$

$$P = \frac{T_P}{T_P + F_P} \tag{20}$$

$$R = \frac{T_p}{T_p + F_n} \tag{21}$$

$$F_1 = 2\left(\frac{P\text{x}R}{P + R}\right) \tag{22}$$

where T_n, T_p, F_n, and F_p refers to true negative, true positive, false negative, and false positive results, respectively.

A. FS Techniques Performance Evaluation

Six FS techniques (extra tree, RF, XGBoost, MI, ANOVA, and LightGBM) were employed to select the best features to improve the classification performance. All ranking-based

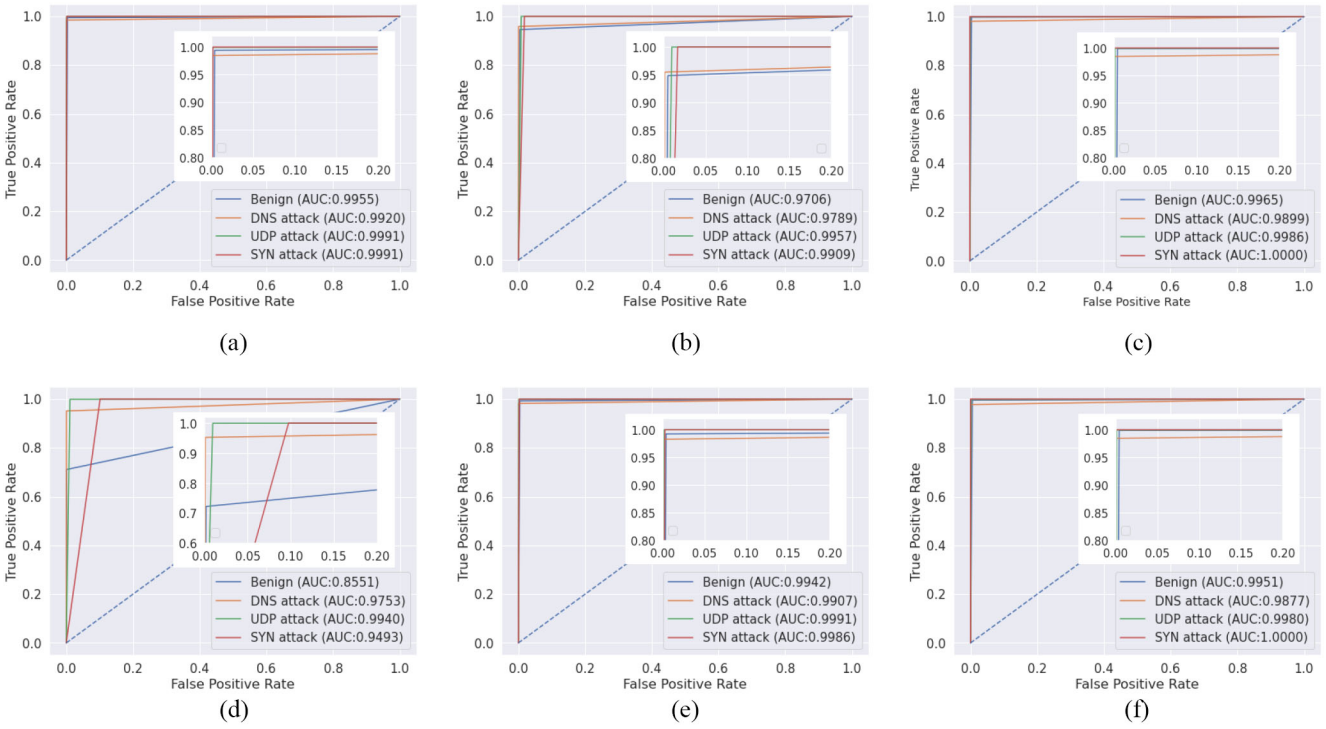

Fig. 7. ROC score for some FS techniques: (a) Extra Trees, (b) RF, (c) XGBoost, (d) MI, (e) ANOVA, and (f) LightGBM for the DDoS classification scenario.

TABLE IV
XGBOOST FS PERFORMANCE WITH DIFFERENT NUMBER OF FEATURES FOR DDOS DETECTION

Performance Matrix	Number of feature				
	f=10	f=15	f=20	f=25	f=30
Accuracy	99.73%	99.70%	99.86%	99.80%	99.83%
Precision	99.63%	99.61%	99.80%	99.73%	99.81%
Recall	99.63%	99.61%	99.80%	99.73%	99.81%
F1-Score	99.63%	99.61%	99.80%	99.73%	99.81%
Loss	0.013	0.011	0.006	0.008	0.007
Time-cost (ms)	0.124	0.251	0.126	0.224	0.250

TABLE V
XGBOOST FS PERFORMANCE WITH DIFFERENT NUMBER OF FEATURES FOR DDOS CLASSIFICATION

Performance Matrix	Number of feature				
	f=10	f=15	f=20	f=25	f=30
Accuracy	99.50%	99.49%	99.67%	99.61%	99.59%
Precision	99.47%	99.39%	99.42%	99.56%	99.37%
Recall	99.46%	99.39%	99.42%	99.56%	99.36%
F1-Score	99.46%	99.39%	99.42%	99.56%	99.36%
Loss	0.022	0.021	0.015	0.018	0.017
Time-Cost (ms)	0.179	0.184	0.250	0.264	0.436

FSs use the best ten features to evaluate the proposed model. XGBoost consistently exhibited superior performance across 10, 15, 20, 25, and 30 numbers of selected features for DDoS detection and classification scenarios as shown in Tables IV and V, further adjudging from the study by Upadhyay *et al.* [48] on the prowess of the XGBoost algorithm for FS. However, all other FS techniques had a substantive performance with ten features. Hence, the model was evaluated using the ten selected most promising features.

Tables VI and VII show comparative analysis of the performance of the various FS techniques considered for DDoS detection and classification. The comparison was based on ten features for each technique, highlighting the performance in term of accuracy, precision, recall, F1-score, loss, and time cost. This analysis indicated that XGBoost is the best FS mechanism for DDoS detection and classification. The DDoS detection-proposed model outperformed all other FS techniques with an accuracy of 99.73% and precision, recall, and F1-score of 99.47%. The minimum error loss achieved was 0.013, and the lowest time cost was 0.124 ms. As seen in Table VII, XGBoost FS outperformed all other FS techniques for DDoS classification, with an of accuracy 99.50%, precision and recall of 99.47%, and F1-score of 99.46%. It has a minimal error loss of 0.022. In this evaluation, the time cost of each FS algorithm was evaluated. XGBoost FS achieved the lowest time cost of 0.179 ms compared to another FS. The fact that XGBoost outperforms LightGBM in terms of processing time in this instance is exciting. Through tremendous hyperparameter tuning, XGBoost can exceed LightGBM for accuracy and time processing performance in specific classification cases.

Fig. 7 presents receiver operating characteristic (ROC) curve for DDoS classification for each FS technique. Based on this ROC curve, XGBoost CNN-LSTM achieved excellent classification ability compared to other FS techniques with an average ROC score of 0.9969. XGBoost CNN-LSTM also performs well for the classification of each class. The detection rates for benign, DNS, UDP, and SYN attacks were 0.9973, 0.9915, 0.9989, and 0.9999, respectively. Because application-layer DDoS attacks are more difficult to detect, DNS attacks

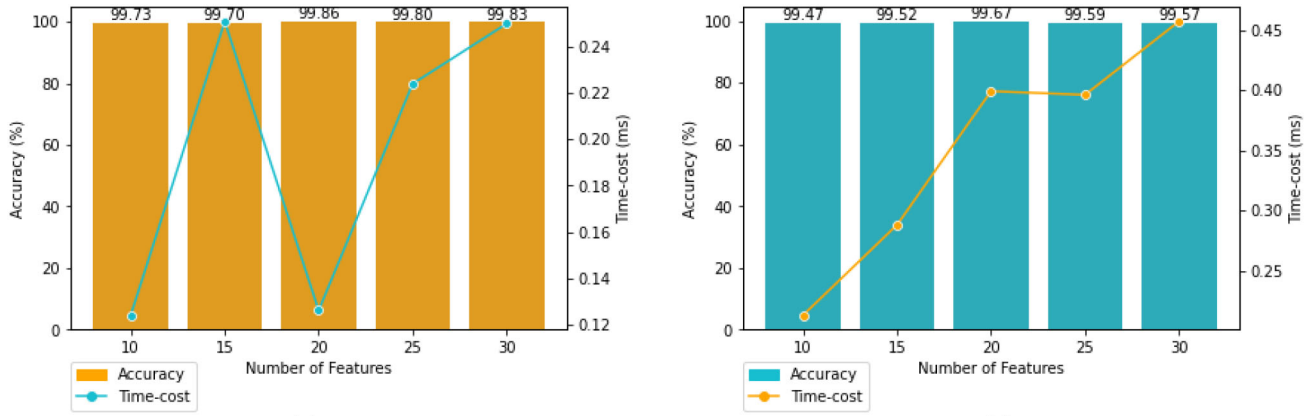

Fig. 8. XGBoost FS accuracy and time cost with different number of features for (a) DDoS detection and (b) DDoS classification.

TABLE VI
FS PERFORMANCE FOR DDOS DETECTION

Feature Selection Technique	Number of Features	Accuracy	Precison	Recall	F1-Score	Loss	Average ROC Score	Time-Cost (ms)
Extra Trees	10	99.52%	99.50%	99.50%	99.50%	0.025	99.55%	0.126
Random Forest	10	97.24%	97.22%	97.22%	97.22%	0.088	97.08%	0.250
XGBoost	**10**	**99.73%**	**99.63%**	**99.63%**	**99.63%**	**0.013**	**99.72%**	**0.124**
Mutual Information	10	85.72%	85.37%	85.27%	85.32%	0.218	85.61%	0.162
ANOVA	10	99.43%	99.39%	99.39%	99.39%	0.028	99.45%	0.127
LightGBM	10	99.68%	99.56%	99.56%	99.56%	0.014	99.69%	0.254

TABLE VII
FS PERFORMANCE FOR DDOS CLASSIFICATION

Feature Selection Technique	Number of Features	Accuracy	Precison	Recall	F1-Score	Loss	Average ROC Score	Time-Cost (ms)
Extra Trees	10	99.38%	99.24%	99.24%	99.24%	0.032	99.62%	0.160
Random Forest	10	97.42%	97.54%	97.53%	97.53%	0.095	98.84%	0.324
XGBoost	**10**	**99.50%**	**99.47%**	**99.46%**	**99.46%**	**0.022**	**99.69%**	**0.179**
Mutual Information	10	91.66%	91.32%	91.30%	91.31%	0.218	94.55%	0.213
ANOVA	10	99.33%	99.29%	99.29%	99.29%	0.034	99.54%	0.214
LightGBM	10	99.40%	99.40%	99.40%	99.40%	0.025	99.55%	0.182

achieved a lower detection rate than other transport layer DDoS attacks (UDP and SYN attacks).

The graph of the accuracy versus time cost of the XGBoost FS approach is shown in Fig. 8. The performance of XGBoost FS for relationship analysis by increasing the number of features and their effect on the training time costs is shown in Fig. 8. Fig. 8(a) shows the best accuracy of 99.86% when using 20 selected features. However, the time cost is high at 0.126 ms, compare to the ten selected features for DDoS detection. Furthermore, XGBoost achieved the best accuracy when exploiting twenty selected features for the DDoS classification. The accuracy was 99.61%; however, the high cost-time required was approximately 0.250 ms. For further measurements, ten significant features were selected owing to high accuracy and low computation time; thus, they can be applied to DDoS detection and classification in a real time.

B. Performance Evaluation of the Proposed Model

Table VIII lists the precision, recall, and F1-scores performance for each class using the proposed model. The proposed model can classify SYN attacks with high precision, recall, and F1-score values of 100%, 99.97%, and 99.99%, respectively. However, the classification ability of DNS attacks requires improvement. DNS attacks achieved low precision and F1-score compared to other classes. The benign group also recorded low recall and F1-score, which requires improvement. The confusion matrix of the proposed CNN-LSTM for DDoS detection and classification is shown in Fig. 9, which provides an error matrix between the predicted and the actual classes. From the results, the proposed CNN-LSTM model can efficiently detect and classify DDoS attacks without confusion. The proposed model can detect all 7214 SYN samples accurately. However, the distributed and reflected DNS is a more sophisticated high-volume application-layer attack. The

TABLE VIII
PRECISION, RECALL, AND F1-SCORE FOR EACH CLASS USING THE PROPOSED MODEL

Class	Precision	Recall	F1-Score
Benign	99.85%	98.67%	99.26%
DNS attack	98.37%	99.73%	99.05%
UDP attack	99.85%	99.73%	99.79%
SYN attack	100.00%	99.97%	99.99%

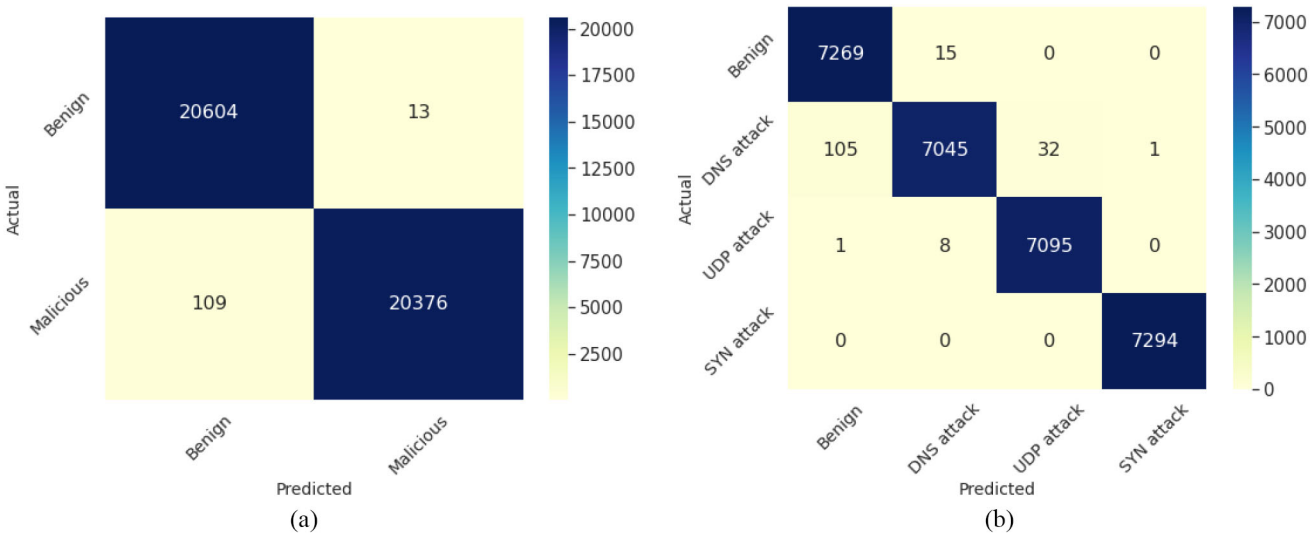

Fig. 9. Confusion matrix of the proposed model for (a) DDoS detection and (b) DDoS classification.

TABLE IX
COMPARISON OF THE REPRESENTATIVE ML AND DL ALGORITHMS FOR DDOS CLASSIFICATION

Techniques	Accuracy	Time-Cost	Learnable Parameter	MFLOPs
Extended Decision Tree [49]	97.00%	-	-	-
CART [13]	98.70%	12.40 ms	-	-
CNN [9]	99.43%	0.155 ms	45.33K	0.828
Deep CNN [24]	99.44%	0.240 ms	34.53K	0.758
GRU [9]	99.43%	0.414 ms	126.49K	5.24
LSTM [9]	99.05%	0.355 ms	55.72K	2.31
Existing CNN-LSTM [25]	99.44%	0.218 ms	32.81K	0.520
Proposed (CNN-LSTM) Model	**99.50%**	**0.179 ms**	**4.83K**	**0.067**

proposed model is still faced with the problem of classifying between benign traffic networks and the characteristics of DNS attacks.

For reliability, the proposed model was compared with the representative ML and DL model structures for DDoS classification, as shown in Table IX. Two ML-based and five DL-based DDoS detection and classification approaches exist: extended decision tree, CART, and DL-based methods, such as CNN, Deep CNN, LSTM, GRU, and the existing CNN-LSTM. We reran the network structure of the existing DL-based approaches for comparison to obtain the accuracy, computational time, the network size (as the number of learnable parameters), and the memory computations (via Mega Floating Point Operations per second (MFLOPs)-Mega Floating Point Operations) using the same network configurations as the proposed model. According to the measurement results, CNN, LSTM, GRU, and the existing CNN-LSTM achieved accuracies of 99.43%, 99,43%, 99.05%, and 99.44%, respectively. The existing CNN-LSTM performed with a time cost of 0.218 ms and a learnable parameter of 32.81K. The proposed model outperformed all other state-of-the-art DDoS detection and classification models with the lowest learnable parameter of 4.83K, time cost of 0.179 ms, and the highest accuracy of 99.50%.

As a consequence, the memory measurement in terms of MFLOPs based on [14] is listed in Table IX as evidence for the computation complexity of the proposed and existing models. As shown in Table IX, the MFLOPs of the GRU, LSTM, CNN [9], Deep CNN [24], Existing CNN-LSTM [25], and the proposed model are 5.24, 2.31, 0.828, 0.758, 0.520, and 0.067, respectively. Although the GRU, LSTM, and CNN [9] achieved good DDoS attack detection accuracy, the MFLOPs and trainable parameters of these models are very high. The high computation complexity of these models can be explained as follows: The GRU and LSTM networks have a high number of hidden neurons to process the input directly. The CNN [9] network cascaded a high-dimensional convolution layer with a large number of kernels without using pooling operations. Moreover, the aforementioned models use a multiple numbers of fully connected layers. Therefore, these models have a huge number of weights with high MFLOPs. However, the proposed model has the lowest MFLOPs (0.067), which is evidence of the least memory consuming compared to the existing DDOS attack detection models and can be easily deployed to the low SDN controller resource consumption.

V. CONCLUSION

This study presents an XGBoost-based FS scheme with a hybrid DL classification method (CNN-LSTM) for software-defined industrial networks. The architecture of the proposed model comprises of three sections: 1) data preprocessing; 2) FS; and 3) attack classification. Primarily, the structure begins with data preprocessing, features are mapped and

ordered to a particular allotment. Subsequently, the FS process followed; the CICDDoS2019 data set was subjected to six FS algorithms to ascertain the data subset with the most promising features. This process had with 10, 15, 20, 25, and 30 data subsets. Stability in the performance of all the compared models was evident in the subsets of ten data features; XGBoost consistently outperformed all other compared algorithms in the considered performance evaluation metrics. This method improved the learning competence. Moreover, the choice of features is dynamically in line with the network traffic. Subsequently, the selected subset of data sets was applied to the CNN-LSTM to classify diverse attack categories.

This study evaluated an efficient DL model (CNN-LSTM) for DDoS classification in IIoT networks using the XGBoost-based FS technique. The model efficiently classified network traffic with a few reliable subsets of the data set and identified various attacks, such as DNS, UDP, and SYN attacks. Furthermore, the proposed CNN-LSTM model utilized residual connectivity and inception networks to obtain a feature-rich and low-complexity network model. The proposed model leveraging the CICDDoS2019 data set is robust and reliable for different numbers of features. Based on experimental results, the proposed CNN-LSTM outperformed the existing approach with an accuracy of 99.50%, a time cost of 0.179 ms, and learnable parameter only 4.83K with MFLOPs 0.067. The viability of the proposed model was validated for a real-time IoT scenario.

REFERENCES

[1] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial Internet of Things: Architecture, advances and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2462–2488, 4th Quart., 2020.

[2] V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107739.

[3] M. Du and K. Wang, "An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 648–657, Jan. 2020.

[4] J. L. Romero-Gázquez and M. Bueno-Delgado, "Software architecture solution based on SDN for an industrial IoT scenario," *Wireless Commun. Mobile Comput.*, vol. 2018, Sep. 2018, Art. no. 2946575.

[5] A. A. Pranata, T. S. Jun, and D. S. Kim, "Overhead reduction scheme for SDN-based data center networks," *Comput. Stand. Interfaces*, vol. 63, pp. 1–15, Mar. 2019.

[6] W. Mao, Z. Zhao, Z. Chang, G. Min, and W. Gao, "Energy-efficient industrial Internet of Things: Overview and open issues," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7225–7237, Nov. 2021.

[7] D. Mourtzis, K. Angelopoulos, and V. Zogopoulos, "Mapping vulnerabilities in the industrial Internet of Things landscape," *Procedia CIRP*, vol. 84, pp. 265–270, Jan. 2019.

[8] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D.-S. Kim, "Composite and efficient DDoS attack detection framework for B5G networks," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107871.

[9] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning," *IEEE Access*, vol. 9, pp. 108495–108512, 2021.

[10] X. Jing, Z. Yan, and W. Pedrycz, "Security data collection and data analytics in the Internet: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 586–618, 1st Quart., 2018.

[11] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del Rincón, and D. Siracusa, "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 876–889, Jun. 2020.

[12] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A hybrid deep learning approach for DDoS detection and classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021.

[13] A. O. Sangodoyin, M. O. Akinsolu, P. Pillai, and V. Grout, "Detection and classification of DDoS flooding attacks on software-defined networks: A case study for the application of machine learning," *IEEE Access*, vol. 9, pp. 122495–122508, 2021.

[14] R. Akter, V.-S. Doan, T. Huynh-The, and D.-S. Kim, "RFDOA-net: An efficient ConvNet for RF-based DOA estimation in UAV surveillance systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12209–12214, Nov. 2021.

[15] M. Nijim, H. Albataineh, M. Khan, and D. Rao, "FastDetict: A data mining engine for predecting and preventing DDoS attacks," in *Proc. IEEE Int. Symp. Technol. Homeland Security (HST)*, 2017, pp. 1–5.

[16] P. Maity, S. Saxena, S. Srivastava, K. S. Sahoo, A. K. Pradhan, and N. Kumar, "An effective probabilistic technique for DDoS detection in OpenFlow controller," *IEEE Syst. J.*, vol. 16, no. 1, pp. 1345–1354, Mar. 2022.

[17] R. J. Alzahrani and A. Alzahrani, "Survey of traffic classification solution in IoT networks," *Int. J. Comput. Appl.*, vol. 183, no. 9, pp. 37–45, 2021.

[18] L. A. C. Ahakonye, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Efficient classification of enciphered SCADA network traffic in smart factory using decision tree algorithm," *IEEE Access*, vol. 9, pp. 154892–154901, 2021.

[19] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021.

[20] S. Haider *et al.*, "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 53972–53983, 2020.

[21] D. Alghazzawi, O. Bamasaq, H. Ullah, and M. Z. Asghar, "Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection," *Appl. Sci.*, vol. 11, no. 24, 2021, Art. no. 11634.

[22] M. Lopez-Martin, A. Sanchez-Esguevillas, J. I. Arribas, and B. Carro, "Network intrusion detection based on extended RBF neural network with offline reinforcement learning," *IEEE Access*, vol. 9, pp. 153153–153170, 2021.

[23] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, Q.-V. Pham, and N.-N. Dao, "A LSTM-FCNN based multi-class intrusion detection using scalable framework," *Comput. Elect. Eng.*, vol. 99, Apr. 2022, Art. no. 107720.

[24] A. R. Shaaban, E. Abd-Elwanis, and M. Hussein, "DDoS attack detection and classification via convolutional neural network (CNN)," in *Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, 2019, pp. 233–238.

[25] L. Karanam, K. K. Pattanaik, and R. Aldmour, "Intrusion detection mechanism for large scale networks using CNN-LSTM," in *Proc. 13th Int. Conf. Develop. Syst. Eng. (DeSE)*, 2020, pp. 323–328.

[26] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Security Technol. (ICCST)*, 2019, pp. 1–8.

[27] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor traffic using time based features," in *Proc. ICISSP*, 2017, pp. 253–262.

[28] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowl. Based Syst.*, vol. 189, Feb. 2020, Art. no. 105124.

[29] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102767.

[30] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *J. Inf. Security Appl.*, vol. 58, May 2021, Art. no. 102804.

[31] Q. Tian, C. Guang, C. Wenchao, and W. Si, "A lightweight residual networks framework for DDoS attack classification based on federated learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6.

[32] I. O. Lopes, D. Zou, F. A. Ruambo, S. Akbar, and B. Yuan, "Towards effective detection of recent DDoS attacks: A deep learning approach," *Security Commun. Netw.*, vol. 2021, Nov. 2021, Art. no. 5710028.

[33] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[34] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, pp. 165263–165284, 2020.
[35] A. K. Sarica and P. Angin, "A novel SDN dataset for intrusion detection in IoT networks," in *Proc. 16th Int. Conf. Netw. Serv. Manag. (CNSM)*, 2020, pp. 1–5.
[36] D. Javeed, T. Gao, and M. T. Khan, "SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT," *Electronics*, vol. 10, no. 8, p. 918, 2021.
[37] H. A. Alamri and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," *IEEE Access*, vol. 8, pp. 194269–194288, 2020.
[38] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, "Intrusion detection in SDN-based networks: Deep recurrent neural network approach," in *Deep Learning Applications for Cyber Security*. Cham, Switzerland: Springer, 2019, pp. 175–195.
[39] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9552–9562, Oct. 2020.
[40] S. Saha, A. T. Priyoti, A. Sharma, and A. Haque, "Towards an optimal feature selection method for AI-based DDoS detection system," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2022, pp. 425–428.
[41] J. L. Leevy, J. Hancock, R. Zuech, and T. M. Khoshgoftaar, "Detecting cybersecurity attacks using different network features with LightGBM and XGBoost learners," in *Proc. IEEE 2nd Int. Conf. Cogn. Mach. Intell. (CogMI)*, 2020, pp. 190–197.
[42] V. S. Desdhanty and Z. Rustam, "Liver cancer classification using random forest and extreme gradient boosting (XGBoost) with genetic algorithm as feature selection," in *Proc. Int. Conf. Decis. Aid Sci. Appl. (DASA)*, 2021, pp. 716–719.
[43] R. Duan, Y. Li, B. Qiang, and L. Zhou, "A feature selection-based XGBoost model for fault prediction," in *Proc. 17th Int. Conf. Comput. Intell. Security (CIS)*, 2021, pp. 232–236.
[44] M. Al-Sarem, F. Saeed, E. H. Alkhammash, and N. S. Alghamdi, "An aggregated mutual information based feature selection with machine learning methods for enhancing IoT Botnet attack detection," *Sensors*, vol. 22, no. 1, p. 185, 2022.
[45] O. R. Sanchez, M. Repetto, A. Carrega, R. Bolla, and J. F. Pajo, "Feature selection evaluation towards a lightweight deep learning DDoS detector," in *Proc. IEEE Int. Conf. Commun. (ICC*, 2021, pp. 1–6.
[46] R. Akter, V.-S. Doan, J.-M. Lee, and D.-S. Kim, "CNN-SSDI: Convolution neural network inspired surveillance system for UAVs detection and identification," *Comput. Netw.*, vol. 201, Dec. 2021, Art. no. 108519.
[47] M. Golam, R. Akter, J.-M. Lee, and D.-S. Kim, "A long short-term memory-based solar irradiance prediction scheme using meteorological data," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, Sep. 2022.
[48] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Intrusion detection in SCADA based power grids: Recursive feature elimination model with majority vote ensemble algorithm," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2559–2574, Jul.–Sep. 2021.
[49] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "Towards effective network intrusion detection: From concept to creation on azure cloud," *IEEE Access*, vol. 9, pp. 19723–19742, 2021.

Ahmad Zainudin received the B.Eng. and M.Eng. degrees in telecommunication engineering and electrical engineering from the Electronic Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree in electronic engineering with the Kumoh National Institute of Technology, Gumi, South Korea.

He has been a full-time Researcher with Networked Systems Laboratory, Kumoh National Institute of Technology since September 2021. He joined the Division of Telecommunication Engineering, Department of Electrical Engineering, Electronics Engineering Polytechnic Institute of Surabaya, as a Lecturer, in 2012. His research interests include intrusion detection system, deep learning, federated learning, and industrial IoT vulnerabilities.

Love Allen Chijioke Ahakonye (Member, IEEE) received the B.Sc. degree in mathematics/computer science from the University of Port Harcourt, Choba, Nigeria, in 2001, and the M.Sc. degree in information technology from the Federal University of Technology, Gaga, Nigeria, in 2016. She is currently pursuing the Ph.D. degree with the Kumoh National Institute of Technology, Gumi, South Korea.

She has been a full-time Researcher with Networked Systems Laboratory, IT Convergence Engineering, Kumoh National Institute of Technology since March 2021. She has over a decade of working experience in the Nigerian oil and gas sector as a Network and System Administrator from 2002 to 2016 and in 2017, she briefly worked as a Logistics Superintendent with Nigerian Petroleum Development Company, Abuja, Nigeria, until 2019. Her research interest is in AI-enabled energy clustering algorithms for smart factories and SCADA vulnerabilities and fault detection.

Rubina Akter received the B.Sc. degree from Mawlana Bhashani Science and Technology University, Tangail, Bangladesh, in 2014, the M.Sc. degree in information and communication technology from Jahangirnagar University, Savar Union, Bangladesh, in 2016, and the Ph.D. degree in IT convergence engineering from the Kumoh National Institute of Technology (KIT), Gumi, South Korea, in 2022.

From 2017 to 2019, she worked as a Lecturer with the Department of Computer Science and Engineering, the International University of Business Agriculture and Technology University, Dhaka, Bangladesh. She is currently working as a Postdoctoral Research Fellow with Information and Communications Technology Convergence Research Center, KIT. Her major research interests include low latency and reliability issues in the industrial Internet of Things, drone surveillance systems, deep neural networks, and machine learning.

Dr. Akter received the Best Excellent Thesis Award from the IT Convergence Engineering Department at KIT.

Dong-Seong Kim (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2003.

From 1994 to 2003, he worked as a full-time Researcher with ERC-ACI, Seoul National University, Seoul. From March 2003 to February 2005, he worked as a Postdoctoral Researcher with the Wireless Network Laboratory, School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. From 2007 to 2009, he was a Visiting Professor with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently a Director of KIT Convergence Research Institute and ICT Convergence Research Center (ITRC and NRF Advanced Research Center Program) supported by the Korean Government at Kumoh National Institute of Technology. His current main research interests are real-time IoT and smart platform, industrial wireless control network, networked-embedded system, and Fieldbus.

Dr. Kim is a Senior Member of ACM.

Jae-Min Lee (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2005.

From 2005 to 2014, he was a Senior Engineer with Samsung Electronics, Suwon, South Korea. From 2015 to 2016, he was a Principle Engineer with Samsung Electronics, Suwon. Since 2017, he has been an Assistant Professor with the School of Electronic Engineering and the Department of IT-Convergence Engineering, Kumoh National Institute of Technology, Gumi, Gyeongbuk, South Korea. His current main research interests are industrial wireless control network, performance analysis of wireless networks, and TRIZ.