IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 July 2024, accepted 9 August 2024, date of publication 16 August 2024, date of current version 28 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3445261

RESEARCH ARTICLE

Rule-Based With Machine Learning IDS for DDoS Attack Detection in Cyber-Physical Production Systems (CPPS)

AYAZ HUSSAIN[1], EVA MARÍN TORDERA[1], XAVI MASIP-BRUIN[1], AND HELEN C. LELIGOU[2]
[1]CRAAX Laboratory, Universitat Politècnica de Catalunya (UPC), 08800 Vilanova i la Geltrú, Spain
[2]Department of Industrial Design and Production Engineering, University of West Attica, 12243 Aegaleo, Greece

Corresponding author: Ayaz Hussain (ayaz.hussain@upc.edu)

This work was supported in part by European Union's Horizon Europe (PHOENi2X) under Grant 101070586, in part by the Spanish Ministry of Science and Innovation funded by MCIN/AEI/10.13039/501100011033 under Grant PID2021-124463OB-I00, in part by ERDF a way of making Europe, and in part by the Catalan Government under Contract 2021 SGR 00326.

ABSTRACT Recent advancements in communication technology have transformed the way the industrial system works. This digitalization has improved the way of communication between different actors involved in cyber physical production systems (CPPS), such as users, suppliers, and manufacturers, thus making the whole process transparent. The utilization of emerging new technologies in CPPS can cause vulnerable spots that can be exploited by attackers to launch sophisticated distributed denial of service (DDoS) attacks, hence threatening the availability of the production systems. Existing machine learning based intrusion detection systems (IDS) often rely on unrealistic datasets for training and validation, thus missing the crucial testing phase with real-time scenarios. The results generated by the ML models are based on predictions at each flow level and cannot provide summarized information about malicious entities. To address this limitation, this study proposed an efficient IDS system that uses both rule-based detection and ML-based approaches to detect DDoS attacks damaging the infrastructure of CPPS. For training and validation of the system, we use real-time network traffic extracted from a real industrial scenario, referred to as Farm-to-Fork (F2F) supply chain system. Both, attacks and normal traffic were captured, and bidirectional features were extracted through CIC-FLOWMETER. We make use of 8 ML supervised and unsupervised approaches to detect the malicious flows; and then a rule-based detection mechanism is used to calculate the frequency of the malicious flows and to assign different severity levels based on the computed frequency. The overall results show that supervised models outperform unsupervised approaches and achieve an accuracy 99.97% and TPR 99.96%. Overall, the weighted accuracy when tested and deployed in a real-time scenario is around 98.71%. The results prove that the system works better when considering real-time scenarios and provides comprehensive information about the detected results that can be used to take different mitigation actions.

INDEX TERMS CPPS, DDoS attacks, Industry 4.0, IDS solution, machine learning, rule-based detection.

I. INTRODUCTION

The fourth Industrial Revolution (4.0) has transformed factories into smart cyber-physical production systems (CPPS), where products, machines, and humans are interconnected across the whole supply chain. The infrastructure of CPPS leverages the utilization of various emerging technologies, such as the Internet of Things (IoT), cloud computing, edge computing, machine-to-machine (M2M) communication, and artificial intelligence (AI) [1]. These technologies have transformed factories into massively interconnected CPPS, laying the foundation for smart factories where the whole chain, from the user to the production plant, is interconnected. This integration of networking, storage, and computing

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

has enhanced the connectivity between different production processes and results in better product quality, increased productivity, reduced cost, and sustainability.

The use of emerging technologies has also resulted in an increased security threat landscape in the operational environments of CPPS, which is constantly evolving. This integration has resulted in novel and non-negligible security challenges, such as an increase in cyber threats, financial losses, data breaches, operational disturbances, and reputation damage. Moreover, since CPPS consists of different IoT-based systems, sensors, and smart devices from different vendors, the chances of showing additional vulnerabilities become much larger, thus leading to potential risks that malicious actors can exploit. In fact, these vulnerabilities can be exploited by the attacker to disrupt the normal operation of the system and consequently result in severe actions, such as shutting down the production line, compromising the quality of the products and causing damage to different assets that are part of the organization. Then it is desirable to deploy an IDS state-of-the-art solution that can protect the perimeter of the network and also allow the different sub-components of the industrial system to communicate with each other without causing any disturbance.

In this study, we focus particularly on the application of CPPS in specifically within the farm-to-fork (F2F) specific supply chain. The F2F supply chain consists of a complete life cycle from the producers (farmers) to the end user who consumes it. As the food industry becomes more dependent on emerging digital technology, the vulnerability to cyber threats like distributed denial of service (DDoS) [2] attacks increases drastically which disturbs the whole supply chain and results in business interruption. The SecuFood project emphasizes the need and importance of analyzing the threats and vulnerabilities associated with the food and supply chain system [3]. The most common attacks that the food industry faces consist of ransomware and DDoS attacks [4]. In general, the advancement of CPPS amplifies the risk of DDoS attacks in the food industry and thus highlights the need for proactive solutions to maintain the integrity and continuation of the food supply chain. Protecting against these types of attacks is mandatory to ensure reliable communication. They can have undesirable effects on the communication infrastructure. The different stakeholders involved in the supply chain of the food industry will be unable to communicate with each other or will experience delays, which can lead to serious financial losses. The attacker can target the different components involved in the communication infrastructure which are connecting the different stakeholders of the food supply chain.

The DDoS attacks can threaten the availability of production lines and overwhelm services and resources. A sudden increase in network traffic and resource utilization can affect the availability of the communication network, resulting in reduced or even worse no communication between different sub-components of the supply chain system. A DDOS attack aims to obstruct legitimate users from accessing the services and, in the worst-case scenario, results in crashing the services. The attacker can achieve its goals by flooding the service infrastructure with an intense number of packets, which can result in consuming all the network resources. In the origin of the denial-of-service (DOS) attack, the attacker emanates from a single point. It is quite straightforward to block the malicious IP as they originated from a single source, by making use of firewalls [5]. It is necessary to understand the tactics, techniques, and procedures used by the attacker to launch these attacks so that suitable mitigation actions can be taken. Various types of techniques can be used by the attacker to increase the frequency and volume of these attacks, for example, the attacker can make use of several infected devices, which are part of a botnet to target the system.

An attacker can use several methodologies to launch these attacks, such as volume-based attacks, protocol-based attacks, and application layer-based attacks. Volume-based approaches focus on consuming network resources to overload the victims' network bandwidth. Protocol-based attacks focus on exploiting vulnerabilities in the network. The application layer-based attacks disturb the communication based on the TCP/IP protocol stack. The attackers can make use of the new attack's techniques classified as zero-day DDoS by exploiting the vulnerabilities and security breaches that have not been discovered yet. The DynDNS was the largest infrastructure attack that resulted in the denial of important services [6]. Keeping the system up-to-date to avoid vulnerabilities and deploying an optimal IDS solution or firewalls such as pf-Sense, Suricate, and Wazhu, can help to protect against these attacks. In addition, the impact of these DDoS attacks and the related vulnerabilities continues to escalate uncontrollably in the context of large, distributed, and diverse systems like supply chains. This could potentially result in the emergence of new attack patterns. These attacks can cause failures in both software and hardware systems. DDoS attacks pose a significant threat to the food industry and can disrupt the entire communication infrastructure. The food industry supply chain usually has a centralized system to make communication between different stakeholders which make them more volatile towards the infrastructure DDoS attacks. They are particularly susceptible to DDoS attacks, therefore requiring advanced and modern solutions to defend against such targeted attacks. This is where the machine learning plays an important rule as it can learn malicious patterns and can handle the attacks sophisticated nature.

Different types of approaches can be used to handle these attacks which includes IDS, deployment of the firewall and ML based solutions. A classical IDS solution makes use of signature-based detection or rule-based detection, which can be used as protection against known attack patterns. The current advancement into machine learning (ML) has gained popularity, particularly in the domain of anomaly detection, and consequently has shifted the trends towards

the utilization of ML-based IDS. The existing research efforts [7], [8] [9] make use of ML-based techniques to handle sophisticated DDoS attacks targeting the industrial system. Major issues associated with these solutions revolve around the absence of real-time testing of developed techniques in real-time scenarios, training with only benign data, and the utilization of non-realistic attack data for training. They also faced the limitation of using attack traffic from different scenarios and the absence of crucial attack information to take the appropriate actions. Furthermore, there is no information available regarding the usability and practical utilization of these detection systems in a real-time industrial environment. Thus, it is without doubt that examining and providing protection against DDoS attacks targeting the supply chain systems demands advanced and robust solutions that can be trusted and can be used in real-time scenarios.

In this paper, we developed an IDS solution for addressing the challenges associated with CPPS F2F supply chain systems against DDoS attacks. The proposed solution makes use of rule-based detection along with the utilization of ML approaches to protect against DDoS attacks. The CICFLOWMETER was used to extract the statistical features from the network traffic obtained from the F2F supply chain. These features were used to train the ML model to learn patterns about normal and malicious network flows. The ML models perform prediction at each flow level thus resulting in enhanced detection capability. While rule-based detection is used to find the known attack patterns. The utilization of rule-based detection helps in detecting known attack patterns, while ML approaches help in increasing the attack detection capability against complicated attack scenarios. The main objective of using both approaches is twofold: i) to enhance the detection capability of the IDS along with improving the decision-making process through reducing the false positive, and; ii) providing an extra check on the predictions made by ML models to make sure the results can be used to give a complete picture of the network situation. Utilization of ML techniques has achieved the detection of each individual malicious network flow but it results in generating the redundancy of alerts for the network administrator if they are not handled properly. Sometimes they can result in an increased number of false positives.

The proposed methodology combines the advantages associated with rule-based detection and thus complements the detection capability of ML approaches against DDoS attacks. The ML models make use of the detection at each individual flow level and classify the normal and malicious network flows. The rule-based detection further complements these prediction results and tries to reduce the false positive generated by the ML models and provides comprehensive results along with assigning the different severity levels to the network flows based on the frequency of the flows. One main issue associated with the existing solution was that they used non-realistic data to train their system and their validation in real-time was questionable. This issue was overcome in the proposed work through the utilization of the real-time benign and attack traffic from a real-time scenario of the food industry supply chain to ensure that the proposed solution can be used effectively in real-time scenarios. Different standard evaluation matrices were used to check the efficiency of the proposed solution in a real-time manner.

The main contributions of the paper can be summarized as follows:

- We reviewed the state-of-the-art approaches and solutions for handling the attacks in real-time industrial systems. The proposed study tried to overcome the limitations and challenges by leveraging the concepts of both traditional and novel approaches.
- We proposed an IDS system that was trained and validated through a real-time supply chain industrial system. A complete pipeline was developed which consisted of data acquisition, pre-processing, feature selection, model training, attack detection, and control actions.
- The proposed IDS system presented a hybrid approach consisting of both traditional rule-based detection and novel machine-learning approaches. The ML-based detection achieved detection at each flow level while the rule-based detection summarized the detection results.
- We have evaluated the usability of our proposed IDS by deploying it in an industrial scenario F2F supply chain system. An adversarial scenario was developed to validate the efficiency of the IDS system. The system achieved an overall higher detection accuracy for both normal and adversarial scenarios.
- We designed an alert system, providing unified alerts along with recommended suggestions that can be used for attack mitigation.

The structure of the paper is as follows. Section II provides a brief overview of existing solutions and state-of-the-art approaches for the detection of attacks on CPPS, including the studies for selected attacks. Section III presents the proposed IDS solution. This section briefly describes the working of the IDS. Section IV presents the proposed architecture deployment and the simulation-based test that was used for the validation of the trained models. Section V provides a detailed analysis of the performance matrices used for the evaluation of the trained models. Section VI provides a discussion of the overall work. Finally, Section VII gives the conclusion and describes the future work.

II. BACKGROUND AND LITERATURE REVIEW

The supply chain has become more complex and involves different actors and different components. The communication infrastructure is considered to be a be backbone of almost all CPPS specifically in the F2F supply chain where all the actors need to communicate efficiently with each other. It is thus vital to utilize several security measures to ensure that this communication network remains uninterrupted when faced with numerous security threats. The research community has developed a great interest in using intrusion detection systems(IDS) from the perceptive of the CPPS. The IDS

are mostly used in the field of monitoring the network infrastructure to protect the systems against intrusion. IDS constantly collects real-time network data and generates different types of alert messages. The IDS solution can classified into different categories based on their deployment or functionality such as anomaly detection and misuse detection [10], [11]. This detection uses the attack's malicious behavior, such as a database of the attack's signature. They sometimes are considered rule-based detection or signature-based detection approaches [12], [13].

However, the recent popularity of machine learning, when applied to anomaly detection has driven its deployment to enhance the performance of traditional IDS, especially in the detection of spoofing attacks [16]. The ML-IDS solution requires a significant amount of data to train and validate ML models to perform the anomaly detection task. The existing research efforts have utilized several datasets such as CIC-IDS-2018 [17], CIC-IDS-2019 [18], UNSW-NB15 [19], Bot-IoT [20], AWID3 [21], and CICIoT2023 [22] to train machine learning model to achieve the attack detection.

In the existing state-of-the-art, many references may be found built on a mixture of both classical solutions and ML techniques to achieve better attack detection. Different works in the literature have utilized various supervised ML techniques as the core of the IDS, as in [23], where the authors present an IDS based on a multi-layer perceptron neural network for intrusion detection. Also, in [24], the authors propose a support vector machine (SVM) based IDS to detect routing layer attacks in an IoT system. In [25], the authors perform empirical experiments using four ML classifiers named Random Forest, Decision Tree, Multi-layer Perceptron, and SVM to test and evaluate the efficiency and performance of IDS. Few studies rely on using unsupervised algorithms to detect zero-day attacks. This is the case in [26], where the authors compare the performance and computational cost of classification models trained with unsupervised ML techniques: principal components analysis (PCA), isolation forest, one-class SVM, and Auto-encoder for the CIC-IDS-2017 dataset. A mixture of both these supervised and un-supervised machine-based approaches to handle the known and unknown attacks was proposed in [27]. Previously discussed works were based on public datasets to propose different types of ML-based IDS for diverse use cases. The list of public datasets is very large and not limited to the ones mentioned above. The efficiency of these public datasets in the real-time environment is not good enough due to the complex nature of the F2F supply chain system.

Several research efforts have developed ML-based IDS to secure critical industrial infrastructure. Saghezchi et al. [7] has focused on the detection of DDoS attacks happening in smart cyber-physical production systems (CPPSs). Their focus of interest was to protect a real-world semiconductor production factory. They experimented with different supervised and unsupervised machine learning algorithms to evaluate the performance of their detection classifier in the real-time use case of the semiconductor production factory. For the training of the algorithm, they used real-time benign traffic from the use case and attack traces obtained from the public repository. They utilized the NetMate [6] tool to extract the 45 bidirectional features, PCA to reduce the number of features, and various ML models to classify the network traffic as normal or malicious. The experimental results show that the Decision Tree approach achieves an accuracy of 99.99%. The main problem with their methodology was that attack traffic from another scenario was used.

Abosuliman [14] has developed a DDoS-based attack detection strategy for the real-time industry of the semiconductor production factory. The focus of this work was to detect DDoS attacks happening in industry 4.0. The proposed architecture consists of extracting the real-time benign network traffic from the real-time systems. Overall, 45 bidirectional features were extracted from the PCAP file, and a labeled dataset was generated to train the supervised ML models. The feature selection is done with the help of the PCA-BSO algorithm, and the tradeoff between different ML models was achieved. For the supervised ML, they trained SVM, logistic regression (LR), and random forest. For the deep learning architecture, they opted for the convolution network (CNN), long short-term memory (LSTM), and gated recurrent unit (GRU). The performance of these algorithms was tested in a simulation environment. The experimental results concluded that the supervised ML models work better as compared to the deep learning models. They utilized only real-time benign traffic for training but missed the validation of the proposed architecture in a real-time environment.

Uszko et al. [9] has developed a methodology to detect attacks occurring in a 5G network. They make use of both rule-based and ML approaches to detect both the known and emerging nature of the attacks. The proposed methodology consists of packet-based inspection and rule-based modules. They utilized the AWID3 [16] dataset for the evaluation of their proposed methodology. The efficiency of the system results in an accuracy of 98.57% and a precision and recall of greater than 92%. The author's goal was to create an efficient system that can be useful in real-time scenarios. One of the latest datasets, AWID3, was used to train and test the system. The utilization of both rule-based detection and machine learning makes the architecture quite reliable for the detection of both known and unknown attacks occurring in the 5G infrastructure. The main criticism of this methodology is that it was not validated with a real-time 5G infrastructure, and only a public dataset was used for validation purposes. The efficiency of the system in the real-time industry is questionable.

Khan et al. [8] proposed a federated learning-based architecture to secure the supply chain (SC) networks to handle the privacy and security issues associated with the supply chain networks. They make use of the distributed local data training to handle the diverse nature of the supply chain. The proposed system was evaluated by making use of the TON_IoT [28] dataset to evaluate their proposed system and achieve an overall accuracy of 99.33%. One

TABLE 1. Comparison of the state-of-the-art research efforts in the CPPS domain.

Reference	Industrial System	Nature of Data		Methodology		Limitation
		Benign	Attack	Packet-based	Rule Base	
Saghezchi *et al.* [7]	Semiconductor Production Factory	Real Traffic	Traces of DDOS	Yes	No	Attack traffic was used from another scenario, with no real-time testing
Abosuliman [14]	Semiconductor Production Factory	Real Traffic		Yes		No real-time testing and attack info is missing
Uszko *et al.* [9]	5G WLAN infrastructure	AWID3 dataset	AWID3 dataset	Yes	Yes	Testing is done with the AWID3 dataset
Linda *et al.* [15]	SCADA Systems	Simulated data	Simulated data	Yes	No	The real-time testing is missing
Khan *et al.* [8]	Supply chain 4.0	TON IoT dataset	TON IoT dataset	Yes	NO	Testing in real-time is missing
Proposed IDS	Supply chain	Realtime	Realtime	Yes	Yes	Trained and tested with real-time industry data.

issue associated with this system was that the scalability into the real-time SC system was not discussed and evaluation lacks the synchronization issues between multiple servers of the federated learning. In [29] make use of the deep neural network and Decision Tree to detect cyber threats in the industrial control environment caused by the vulnerabilities associated with the integration of the IoT systems. The proposed system achieves better detection results as compared to conventional classifiers but real-time testing in the industrial environment was missing. The [30] utilized a deep learning-based IDS system to protect the SCADA networks. The IDS was designed to protect the networks from conventional and domain-specific attacks that can target the SCADA networks. The experimental results show that KNN and RF achieved higher accuracy. They achieved an overall accuracy of 99.75% in the detection of the attacks.

Various solutions and methodologies were developed to handle the sophisticated nature of the attack occurring in the industrial scenarios. The [31] developed 4 stage methodologies to check the efficiency of the anomaly detection models in industrial scenarios. They make use of the deep learning architecture Long-short term memory (LSTM) and 1-dimensional deep neural network(1D-CNN) to detect the anomalies and experimental results in the testbed showed that 1D-CNN is more robust as compared to other architecture. Wang et al. [32] presented a transformer-based architecture to predict multi-stage attacks. They utilized the concept of alert aggregation and specifically predicted the stage of the attack. For their model's training, they used a testbed to develop the datasets of various alerts being used for training purposes. In [33] an IDS system to handle the man-in-the-middle attacks occurring in the private networks of the smart grid industrial systems. They make use of the real-time data collected from the smart grid to train their models and achieve an accuracy of 97.6% to 100%. Huma et al. [34] proposed a hybrid deep learning architecture for the IoT networks and tested their methodologies with UNSW-ND15 [19] and DS2OS [35] datasets. They achieved a higher accuracy of around 98% and 99% respectively. Presekal et al. [36] utilized the attack graph models to handle the attacks in the power systems. It makes use of the hybrid deep learning models along with the Graph convolution LSTM to achieve early attack detection. The experimental results have shown that the proposed method can identify the active attack's location and achieve an accuracy above 96%. Hu et al. [37] proposed an entropy-based IDS system to detect stealthy attacks on industrial systems. This approach was effective in data modification attacks. In [38] discussed the issues associated with the F2F supply chain system and presented a FISHY platform that can protect the whole supply chain system concerning different threats that can happen in the supply chain network.

In [39], author utilized the physics-based attack detection to protect the CPPS. In [40], utilized the estimation models to protect the industrial system against false data injection and jamming attacks. Dolk et al. [41] designed strategies to enhance the resilience of the event-triggered system for the Denial of the system attack. Amodei et.al. [42] makes use of ML approaches to detect attacks in IoT networks. In [43], the author argued that Once-Class Support Vector Machine (OC-SVM) works better for the detection of anomalies in the SCADA and ICS systems. Linda et al. [15] uses artificial neural networks to detect the different anomalies in the SCADA systems. The dataset used for the training was generated in the test environment. A comparison of the related research efforts towards complex CPPS is summarized in Table 1.

The complexity of the CPPS infrastructure requires that the ML algorithms need to be very precise in the detection of the attacks. Each CPPS consists of a different infrastructure which requires the algorithm to be trained with real-time data. ML-based IDS seems to be a very prominent solution to detect the latest types of attacks in complex systems but still, it seems to be insufficient [44] if it is used alone. These ML models require a large amount of clean data that can be trusted and can be used further for the training. In the literature, few researchers make use of both machine learning and rule-based detection to enhance the decision-making of the IDS which is the base of this proposal. Saghezchi et al. [7] makes use of ML-based packet analysis to train their models. The main

criticism of this research is that the testing in real-time is missing. The architecture proposed by Uszko et al. [9] makes use of both packet-based inspection and rule-based detection to enhance the detection capability of IDS. One major issue with all these research efforts is that their efficiency in the real-time industry needs to be evaluated. These systems were trained and tested with only public datasets or made use of a mixture of real-time and public datasets. There is no evidence found on the utilization of these systems in a real-time environment. There is a strong need to have such an IDS solution that can overcome these limitations.

Few research efforts have proposed domain-specific attacks occurring in industrial environments such as IoT, the Power sector, and semiconductor factories. Mostly they have used self-generated datasets or public datasets from different scenarios which makes it difficult to be applicable in the real world. One major issue with all these research efforts is that their efficiency in the real-time industry needs to be evaluated. These systems were trained and tested with only public datasets or made use of a mixture of real-time and public datasets. There is no evidence found on the utilization of these systems in a real-time environment. There is a strong need to have such an IDS solution that can overcome these limitations.

III. PROPOSED IDS SOLUTION

This section provides the proposed IDS solution for the real-time detection of attacks in supply chain infrastructure. The Proposed solution discusses the steps that will be carried out throughout this paper. The proposed methodology is divided into six key steps: 1) Data Acquisition, 2) Pre-Processing, 3) Feature Selection, 4) Model Training, 5) Attack Detection, and 6) Control Actions. These steps with their inputs and outputs are shown in Figure 1.

A. DATA ACQUISITION

As highlighted previously the first step is data acquisition which involves the collection of data, extracting features from the data, and the transmission of the data to be available for future components. These steps are briefly discussed below.

1) DATA COLLECTION

The data collection task was done by capturing the network traces from the target CPPS. This task can be performed by using tools such as tcpdump, wireshark and tstat can be used to sniff the network packets and can be dumped to a PCAP file. Among these tools, tcpdump was selected because of its open-source nature and widespread use in network flow-capturing tasks. The tool makes use of packet sniffing techniques to passively capture all the data link layers framer passing through a specific network adapter. A shell script was designed to capture the network traffic floating in the targeted infrastructure which serves as a base for the IDS to start its working. This network traffic contains all the network flows consisting of the intended operations in the supply chain. During the sniffing mode, the tool configures the network card to put the packets in a packet queue, later these are copied to a core protocol stack. These PCAP files are captured continually and dumped into PCAP file format which are initially stored locally.

2) FEATURE EXTRACTION

The data collected in the previous step consists of the raw network traffic which needs to be processed to build into the desired specific format. This data is then used for the training of different ML models. Different statistical features can be extracted from this traffic to characterize the network flows. There are two highly advanced tools, namely NetMate [45] and CIC-FLOWMETER [46], that can be employed to extract valuable statistical network features from the network traffic. In this study, the CIC-FLOWMETER was chosen due to its extensive capability of calculating over 80 bidirectional network features from the PCAP files. These statistically computed characteristics hold a significant role in data modeling. The tool generates bidirectional flows by considering the initial received packet and determining the subsequent and reverse directions. Additionally, the CIC-FLOWMETERE· generates network traffic characteristics for both the forwarded and backward flows within the network. These characteristics encompass variables such as duration, number of packets, and number of bytes. Furthermore, it presents six valuable features, including a timestamp, source IP, and destination IP, which provide crucial details about a specific flow. The extracted features were stored in the csv file format.

3) DATA TRANSMISSION

The next step is the transmission of data which is done by deploying the IDS in the distributed environment. The data collection part resides in the lower infrastructure from where it collects the data. Initially, all the extracted features from the CIC-FLOWMETER were stored in the csv format to be transmitted for further processing. A Node-JS-based REST-API was being developed to transmit these csv files to the Kubernetes environment where the rest of the components of the IDS are working. It continually waits for the new request of data and as soon as the request is received, the new csv files containing the statistical features are converted into the JSON format and are transmitted.

B. PRE-PROCESSING

Pre-processing is performed on the data to remove uncertainty. It generally refers to adding, removing, and transforming data to achieve the required data for training purposes. To illustrate the general pre-processing technique, we begin by using duplication removal. The data set contains common predictor problems such as missing values, noisy data, outliers, and other empty labeling-related issues. Below we have described each step of the pre-processing in detail.

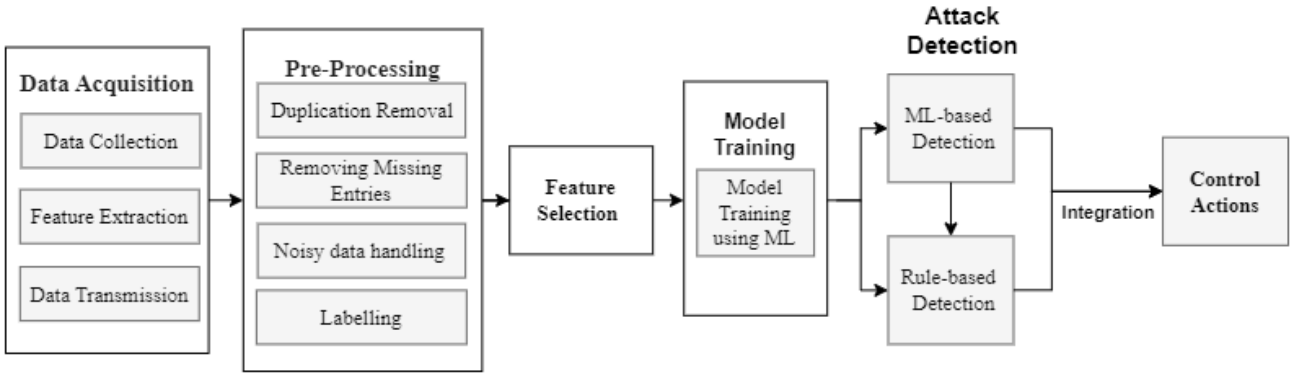


FIGURE 1. Flow of the proposed IDS solution.

1) DUPLICATION REMOVAL

The extracted features need to be processed further so that they can be used for the model training. The CIC-FLOWMETER extracts 83 network statistical features and a blank label column. We have utilized the pandas library available in R to achieve this functionality. The data was loaded into the pandas data frame which allows the easy manipulation of the data and then a built-in function was utilized which finds all the duplicate rows and removes all the duplicate entries accordingly.

2) REMOVING MISSING ENTRIES

The extracted features contain several entries which either contain ''NA'' entries or blank entries. In the first step, we identified the missing entries and converted them into the whole data frame and then removed these entries using the panda's library built-in functions. We have removed all the rows which contain either ''NA'', ''NaN'' or blank entries. This removal leads to more reliable and accurate data analysis.

3) NOISY DATA HANDLING

The collected data contains several discrepancies due to its retrieval from a real-time system. The timestamp field was used as a first criterion to filter out the intended network flow activities. The data can contain several outliers which can impact the training process. The data needs to be normalized to reduce the random variability in the data. The data normalization techniques depend on the data and ML algorithm used for the model training. We utilized the Min-Max scaling to normalize the numeric features. It transforms the features into a specific range of 0 to 1. Normalization of the features along with the type's conversion is performed.

4) LABELLING

Initially extracted features from the network flows contain a label column which is initially empty. For the training the ML model, the data must be labeled properly so that ML models learn the patterns effectively and can perform the prediction in real-time. We have established the two scenarios including attack scenario and normal traffic respectively. Inspired by the data labeling from the article [47] which was being used for the development of the CIC-IDS-2018 dataset. Although this data labelling approach can lead the algorithms to be biased towards the IP address, port address, and protocols can cause overfitting issue. However, to avoid this problem, during the training stage these features were being discarded so that models can learn the normal and attack partners efficiently. Some features such as Bwd PSH Flags, Fwd URG Flags, and Bwd URG Flags remained constants for both the benign and malicious traffic and their values remained zero. Initially, the number of features extracted from the feature extraction block is 83 along with a blank label column. However, following the elimination of features like Flow ID, Src IP, Src Port, Dst IP, Dst Port, and Timestamp, the number of features was reduced to 77. Those attributes that exhibited consistency across the traffic data were subsequently excluded. leaving only the remaining features to be utilized for the purpose of feature selection.

C. FEATURES SELECTION

Initially, the raw network feature extracted using the CICFLOWMETER is 83 statistical features along with the label column. However, not all features are useful for model training. A few features such as Flow ID, Src IP, Src Port, Dst IP, Dst Port, and Timestamp were highly dependent upon the network traffic and environment from where the network traffic being is collected. The aforementioned features do not contribute in identifying the normal and attack network flows, but these features can hold a significant value in identifying the malicious entities. These features were discarded during the training stage to prevent the potential bias in the ML algorithm towards specific ip addresses.

During the feature selection process, the variance of the few features remains consistent such as Bwd PSH Flags, Fwd URG Flags, and Bwd URG Flags remained constants for both the benign and malicious traffic. These features possess a persistent value of zero, leading to their initial exclusion from

the analysis. While selecting the features selection algorithm, we utilized the wrapper methods for the features selection. The wrapper methods, as described by the warren [48], play an important role in selecting the optimal number of features to improve the accuracy of classification accuracy.

We utilized the random search feature selection method which selects the random subsets of the features and evaluates their performance on a batch size five. The classification error was used to optimize the performance of the algorithm. The stopping criteria for each feature was 20 iterations. During the optimization stage the best subset of the features We utilized the random search feature selection method which selects the random subsets of the features and evaluates their performance on a batch size of five. The classification error was used to optimize the performance of the algorithm. The stopping criteria for the feature selection was 20. During the optimization stage, the best subset of the features was constantly updated and at the classification task was then updated to include only selected features. We used 3 fold cross-validation strategies to evaluate the performance of the model. Cross-validation is a robust method to evaluate the performance of the model by splitting the data into k subsets where k is equal to 3 in our scenario. Each fold is used as a validation set while the remaining k-1 folder was used for training the model. By using this approach, we aim to obtain the general performance metrics, mitigate overfitting issues, and ensure the generalizability of the selected features.

D. MODEL TRAINING

The state-of-the-art ML algorithms used by several researchers are reviewed in Section II. These research efforts have utilized several methodologies and resulted in achieving a high accuracy of greater than 90%. It is challenging to determine the most suitable methods which can be useful in real-time industrial scenarios. Unlike previous studies, the proposed study has utilized real-time benign and attack traffic for the training and testing of the model resulting in overcoming the limitation associated with public datasets. Custom datasets obtained from the real-time industrial scenario were used to test and validate the ML model to get more realistic results.

Given these factors, we made use of different ML approaches which were trained to differentiate between the normal and malicious network flow and the best-performing algorithms were tested in real-time. We selected overall 8 supervised ML and unsupervised algorithms to start evaluating the efficiency.

- **Naive Bayes** is a simple probabilistic algorithm that is quite simple but powerful enough to be used in classification-related activities. It can be applied to detect malicious activities by checking the deviation through the normal behavior.
- **Ranger Forest** is a fast implementation of the random forest algorithm. It is quite suitable for the classification task and uses different features to identify normal and malicious flows.
- **Logistic Regression** is a statistical method that is useful for binary classification. It makes use of a sigmoid function to model the probability of the binary outcome.
- **k-Nearest Neighbors (KKNN)**is an extended version of the KNN. The simple KNN algorithm works on calculating the neighboring distance using the Euclidean distance while the KKNN makes use of the Kernal function to calculate the weights of neighbors, resulting in better non-linear relationships between the data points.
- **eXtreme Gradient Boosting (XGBoost)** is quite effective in different classification regression and ranking tasks. It is an ensemble learning algorithm that is quite suitable for structured data.
- **Multinomial Naive Bayes (Multinom)** is an extended version of the naive Bayes.
- **Neural network** is a computational model which consists of multiple layers. The neural network is used in the various domains of cyber security to achieve the task of anomaly detection.
- **One-Class Support Vector Machine (OC-SVM)** algorithm can be used for unsupervised ML to train only on the normal distribution of the data to detect abnormalities in network traffic.

The performance of the algorithm depends upon optimizing the hyper-parameters. These models were trained on the training data obtained from an industrial scenario of supply chain farm-to-fork use case. After the training and validation, the models were deployed in real-time scenarios to check their efficiency. While other complex ML approaches such as convolution neural network (CNN) and other variants of the deep learning approaches as reported in the state-of-the-art can produce remarkable results, they are not explored in this study. The primary rationale for this was that they required a large amount of training datasets with a larger distribution of the attacks.The proposed IDS makes use of both real-time data for the training and validation of the system. The selected models are often faster to train and run as compared to convolution neural networks and other variants of deep learning models. Another reason for opting for using these simple ML models lies in the interpretability. It is easier to understand how the classification decisions are made. Combining these predictions with rule-based detection provides comprehensive results. These decisions can be used for taking some mitigation actions in the real-time scenario.

E. ATTACK DETECTION

Rule-based and machine learning-based attack detection are two promising approaches for the detection of the network's attacks. The rule-based detection uses patterns and predefined rules, indicating specific attacks in the infrastructure. While the ML approach uses statistical algorithms to train and model the algorithms to predict the specific type of network anomaly. A hybrid approach that utilizes both approaches results in combining the advantages associated with both techniques.

1) ML-BASED DETECTION

Advancements in computing resources over the past few decades have shifted the trends of using artificial intelligence in every field. Machine learning has also found its application in the anomaly detection task due to its extensive capability and high-accuracy results. The trained models make use of the statistical algorithms trained on features extracted from the PCAP file obtained from the real-time infrastructure. They provide predictions about each bidirectional flow and assign a predicted label to each flow. A single pcap file can contain a large number of bidirectional flows and each flow is predicted as normal or malicious with an accuracy of greater than 90%. The detection results contain each network flows which are classified as normal or malicious. This detection is not sufficient to take any action by the network administrator. If it is used alone, it can result in triggering a larger number of alerts for a single pcap file if only ML is used to perform the alert generation.

2) RULE-BASED DETECTION

Rule-based detection is a classical approach to defend against known threats. The proposed IDS utilized rule-based detection to complement the detection capability offered by ML models. Rule-based detection helps to take quick actions thus reducing the risk of damage to the network. The ML models give predictions about each bidirectional flow and assign a predicted label to each flow. These prediction results were further enhanced using rule-based detection. A possible DDoS attack rule that can be implemented in Suricata, the traditional detection tool, is as follows:

alert tcp any any ->
HOME_NET80(*msg*: "*Possible DDoS attack*";
flags: *S*; *flow*: *stateless*; *threshold*: *type*'*both*, *track*
by_dst, *count* 200, *seconds*1; *sid*:1000001; *rev*:1;)

The keyword "alert" signifies that an alert will be triggered when the rule's condition is met. The rule specifically targets TCP traffic, ensuring that it originates from the local network. It examines the TCP SYN flag, while the flow being stateless indicates that each flow is monitored. Additionally, it checks if there are 200 packets observed within a 1-second interval to generate an alert. When the DDoS attack is happening in the network, the system will receive a significantly large amount of requests which gives an initial indication about the attack. In this study, our focus was protecting against DDoS attacks targeting the F2F supply chain environment. The attacker can use multiple techniques and approaches to accomplish this task. A similar detection rule implemented in suricate was coded in R which works on the respective features available in each bidirectional network flow. The implemented rule uses volumetric DDoS attack detection approaches and observes the fact that when a DDoS attack is happening, it can originate from multiple sources. It also considers the predictions performed by the ML model and checks what labels were assigned to one particular flow. The Pseudo code is given in Algorithm 1.

Algorithm 1 Pseudo Code for DDoS

```
Initialize a dictionary to keep track of counts per (Src.IP, Dst.IP, Protocol, prediction) tuple
Initialize a dictionary to keep track of last-seen timestamps per tuple
for each row in df3 do
    tuple ← (Src.IP, Dst.IP, Protocol, prediction)
    currenttimestamp ← row.timestamp
    if the tuple is not in the counter then
        counter[tuple] ← 0
        last_seen_timestamp[tuple] ← currenttimestamp
    end if
    if currenttimestamp − last_seen_timestamp[tuple] ≤ 1 second then
        counter[tuple] +=1
    else
        counter[tuple] ← 0
        last_seen_timestamp[tuple] ← currenttimestamp
    end if
end for
```

The provided pseudo code keeps track of the frequency of the flows based on the timestamp. It keeps count of the flows based on IP address, protocol, and prediction done by the model. It starts the counter with zero and assesses whether the difference between the current time and the past stamp is less than or equal to 1 second. In this way, the frequency of specific flows over time is calculated. The higher frequency indicates malicious activity.

3) INTEGRATION OF RULE-BASE DETECTION WITH ML-DETECTION

ML model performs the prediction on each flow, and it assigns a predicted label for every flow. These results are then used by the rule-based detection. The results are loaded into a data frame. First of all, a dictionary is generated to keep count of the occurrence for each unique (ip, protocol, and predicted label) tuple. The last time stamp for each of the unique tuples is stored in another dictionary. The integration of the flows is done based on the timestamp. Given the nature of the DDoS attack, the attacker sends multiple requests for the target services and the timestamp between these requests is very small. A counter has been initialized which keeps track of the current timestamp and last seen time stamp, if the difference between them lies within the time window, then it increments the counter and if it is outside then the counter is again reset. The time window size in this scenario is considered 1 second and can be adjusted depending upon the use case. Through using this approach, the prediction done by the ML detection is summarized and similar flows are generated through this approach. Further, the frequency of the network flows was calculated to trigger the alerts.

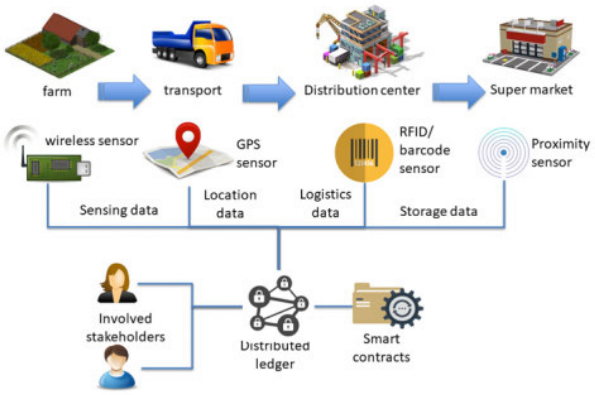


FIGURE 2. An overview of the Agri-food industry [49].

F. CONTROL ACTIONS

The final results obtained from the IDS system consist of the unique network flows along with their frequency of being predicted as normal or attack. Alert messages are further enhanced, and classified into assigning different severity levels (High, Medium, Low) based on the total traffic contribution which gives a useful insight into the IP address which were predicted to be involved in the attack. The threshold for triggering an alert for taking an action is based on the contribution of the traffic accommodated by the top malicious network flows. For the sake of experimentation, as it was done in a real-time environment, only network flow that has High severity levels was being used to generate an alert. The recommended actions to be taken are derived from the MITRE Framework, which offers a range of recommendations based on the characteristics of the attacks. The framework proposes appropriate actions tailored to the specific nature of the attack. To effectively mitigate DDoS attacks, it is advisable to either block malicious IP addresses or the ports that are anticipated to be malicious. This list of recommended suggestions has been shared with the system administrator, who can then implement these actions within the infrastructure.

IV. DEPLOYMENT

The Proposed IDS was tested and validated using the real-time scenario of a food-based industry, a farm-to-fork (F2F) supply chain management system.

Fig. 2 gives an overview of the food industry life cycle which forms complex ICT scenarios [49]. The food industry life cycle is complex and consists of many actors and stakeholders responsible for performing different functionalities. This life cycle consists of all stages from the production of the food, transportation, distribution and then reaching the supermarkets where it is consumed by the end users. It can contain a large amount of information related to food which is produced during different stages of this supply chain. This information can be stored in the distributed ledger where the information remains available for the consumer.

A. SUPPLY CHAIN INFRASTRUCTURE

The IDS system needs to be trained and then validated with real-time industry scenarios. The perceptive working of the proposed IDS into F2F is shown in Fig. 3. This complex system involves different threat actors, and it is complicated to handle all the threats that are associated with these actors. In the F2F supply chain, the data is collected from different IoT sensors and IoT islands and stored in blockchain-based ledgers. The F2F supply chain has a SOFIE web-based application which is being used as an interface between different actors. It acts as a backbone to interact with the Farmer, Transporter, housekeeper, and consumer to insert and see different information related to the food life cycle. It is an interface that is being used by the users and IT solution to interact with the whole supply chain. Apart from several research challenges associated with this complex supply chain system, it is necessary to protect this SOFIE platform so that the supply chain normal operations are not disturbed. This platform is the backbone of the supply chain, and it is necessary to provide security measures for both inside and outside threats that can disturb normal operations. One of the major threats associated with this system is that the attacker tried to put the system in such a state that it became inaccessible to the rest of the legitimate actors who wanted to perform genuine operations. The Goal of the proposed IDS solution is to defend the systems against this type of internal and external threats and notify the administrator about the malicious network flows that are trying to perform some nasty stuff. The IDS system should monitor the network traffic in real-time and alert the network administrator against malicious activity. To start its work, it first needs to be trained and then deployed in the infrastructure to start its working.

B. DATA ACQUISITION FROM INDUSTRIAL SCENARIO

The proposed IDS requires real-time data for training, which must be extracted from the real-time environment. In this implementation, we opted for a distributed deployment, where data collection is centralized in the lower infrastructure, while the rest of the processing modules were working in a Kubernetes cluster. The training data was being gathered from the F2F infrastructure. This data consists of all the operations intended for the SOFIE web application. This web application interacts with all the actors which encompassed a supply chain system. The data acquisition was done by using the tcpdump tool which captured all the incoming and outgoing traffic intended towards the SOFIE platform. To start the training the ML models training, we first collected the normal traffic which was collected. This data was collected from the real-time operation of the system where the actors were performing intended operations in the supply chain such as adding new data about food, shipping information by transporter, logistic information, and then markets where the food is being stored. To capture all the needed info and have all the normal traffic partners, the normal operation consists of two states of the system.

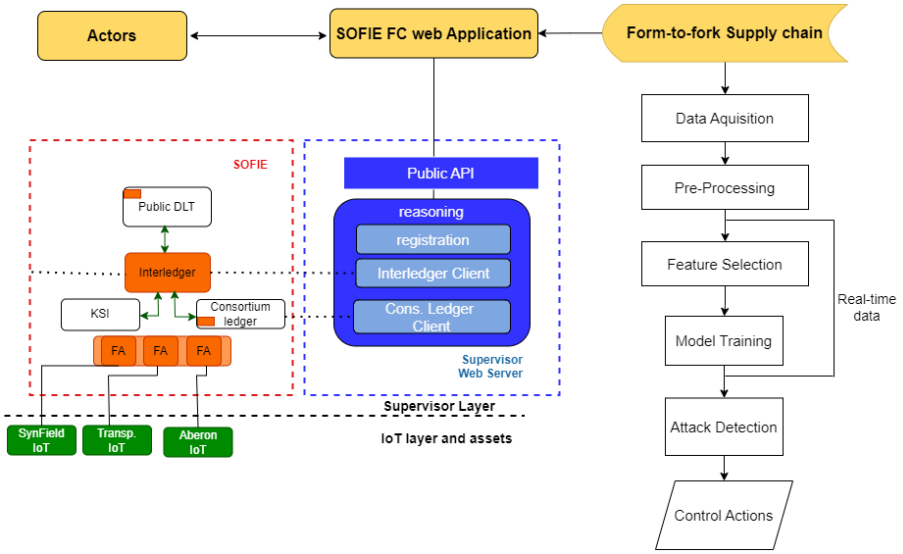


FIGURE 3. A workflow of the proposed IDS concerning the farm to fork supply chain.

TABLE 2. Overview of the training data.

Type of Data	Timestamp	Data Size	No. of Features	Total. No of Flows	Duration	Label
Normal operation	2023-05-24-13 to 2023-05-29-13	166Mb	83	80000	5 Days	Normal
DDoS attack traffic	2023-06-20	158Mb	83	30000	135 mints	Malicious

Initially, the system's normal operation was captured during the idle state, where no specific activity was being carried out on the SOFIE. During this duration still actors were interacting with the system and this traffic was also being captured. Subsequently, during the busy state, we deliberately performed different legitimate operations that could be conducted by the farmer, transporter, housekeeper, and consumer. All these operations were collected, captured and dumped using the data acquisition block of the IDS. For the training purpose, a total of five days of normal operation data was used. Similarly, we simulated the different types of DDoS attacks using different types of approaches. we conducted DDoS attacks at different intervals and during different time slots. We designed different attack scenarios because of the threats towards SOFIE platform. We examined a scenario where an internal machine was infected and exploited by the attacker to initiate a DDoS attack through a botnet. The adversary gained access to an internal machine as part of the infrastructure by exploiting some vulnerability. Additionally, we considered a scenario where the requests were coming through the public router. The DDoS attacks were generated using various techniques. For instance, the adversary attempted to send illegitimate requests to different system services to disrupt normal operations and degrade system performance. In the worst-case scenario, if the attack is successful, the system may enter a state of denial of service. The traffic that was directed towards SOFIE, as depicted in Fig. 3, was captured and further analyzed. When generating the attacks, we intentionally simulated them from within the organization's networks, using known IP addresses that were part of the botnet. These IP addresses were utilized to validate the detection capability of the IDS. We simulated attacks for different intervals of time. A sample of the data is shown in Table 2.

There were a few limitation constraints associated with this traffic. The normal operation of the system tried to be captured for five days but it might not have all the operations intended for the system and can result in complications. The data might contain simulation artifacts as the attacks were generated using different attack approaches through simulation approaches.

The Flow of the IDS in real-time is as follows:

Data Aquisition: The data from the F2F was collected using tcpdump and stored in the PCAP files which were later processed by the CICFLOWMETER to extract the features. It extracted all the statistical features for the bidirectional network flows. These extracted features are stored in the csv format which later needs to be transmitted. In a real-time manner, only features are transmitted rather than PCAP files. It is harder and time-consuming to transmit the PCAP file so only extracted features are transmitted which results in better detection and quick response time.

This extracted csv file was then transmitted through using the REST API to the Kubernetes environment where the rest of the IDS modules are working. These functionalities were accomplished through data collection,

feature extraction, and data transmission blocks of the IDS.

C. REALTIME WORKING IN INDUSTRIAL SCENARIO

The training data gathered from the previous step was then used to start training the machine learning models.

1) PRE-PROCESSING

step of the IDS uses the training data to start transforming this data into a useful form so that we can use it in the next stage. Initially, the duplicate entries and NA entries are removed from the training data. The training data needs to be balanced to avoid the model overfitting. The data was normalized to minimize the effect of the noise. The data labeling was being done considering the duration when it was being captured. Data labeling was based on the Timestamp fields and considered the duration of normal system operations. The same process was repeated for the traffic attack. Labeling was based on the duration of time during which the attacks were simulated.

2) FEATURE SELECTION

stage of the IDS is responsible for selecting the optimal features which can be then used to train the models. To remove the biasness, features such as ip address and ports are removed so that the model can be trained adequately and learns rather than becoming biased towards some specific ip address. The most valuable features obtained from the training data were then used to train the ML model.

3) MODEL TRAINING

stage uses this training data to start training the ML model. Only 80% of the training data was used for the model training and the rest of the 20% of the data was used for testing purposes. For the validation purpose again, real-time data from the infrastructure was used to check that models are working correctly. Out of all the trained models, only best best-performing models were used in the deployment stage where they were validated with new data.

D. SIMULATION-BASED EXPERIMENT

The proposed IDS solution was tested and evaluated through a simulation-based experiment. The model trained through training data was then used to validate their effectiveness after fetching new data from the F2F. For validation purposes in real-time scenarios, new data is acquired through Data Acquisition which is followed by the pre-processing step and then they are followed by attack detection.

Attack Detection: serving as the core of the IDS, utilized the trained ML model to make predictions for new data and assign predicted labels to each flow. The rule-based detection mechanism further refined these predictions, while the Recommendation/Alert unit proposed suitable actions and displayed the outcomes on a dashboard of IDS accompanied by various statistical features that offered deeper insights into network traffic.

TABLE 3. Comprehensive results during simulation experiments.

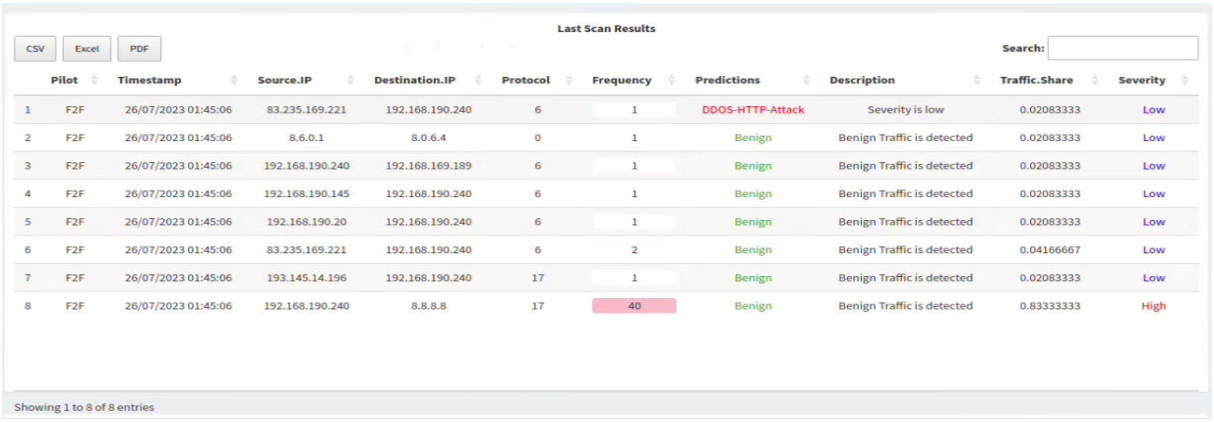
Scenario	DR	Precision	Recall	F1 Score
Normal Activity	97.87%	97.87%	100%	98.92%
DDoS Attack	98.20%	100%	98.20%	99.10%

1) ADVERSARIAL SCENARIOS

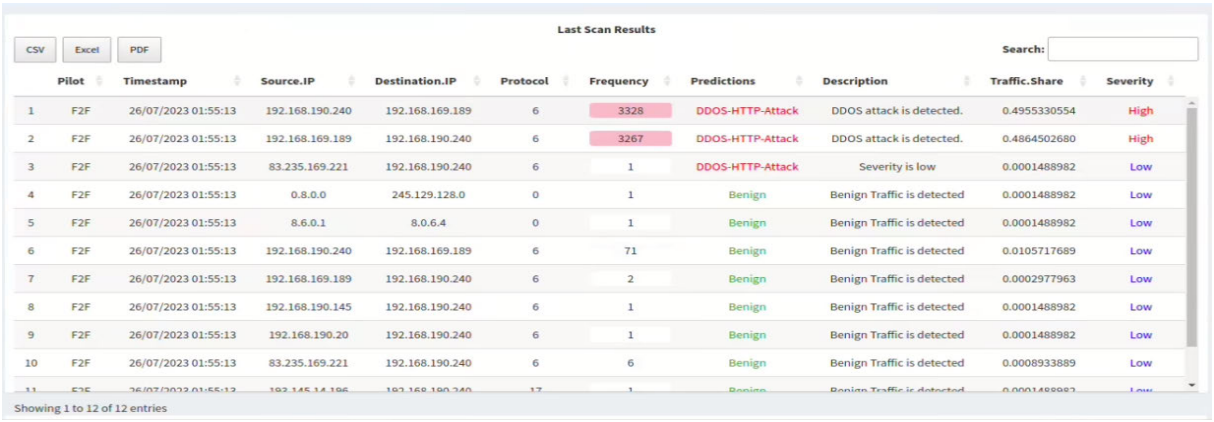
To check the effectiveness of the IDS system, we designed adversarial and normal scenarios to verify how the system behaves when validated with new data. First, we simulated the normal operation of the system. During the normal operation, we simulated supply chain operation for known Timestamps. These actions were captured and transmitted to the attack detection block automatically in real-time. We assumed that during this timestamp no malicious activity is performed. It was expected that the attack detection module which uses trained ML models followed by rule-based detection should classify and mark these activities as normal. The trained model performed predictions for all flows and the results were followed by rule-based detection which further summarized the results. These integrated results are then used by the control actions which can be used by the administrator to take some mitigation actions.

The detection results under normal traffic conditions are visualized in Fig. 4(a). As observed, the trained model accurately predicted the system's normal behavior, correctly categorizing the flows as benign. The dashboard provided statistical information about the network's different flows, along with useful statistics concerning traffic share and flow severity based on their respective shares. Notably, certain DNS resolution queries were captured alongside the normal flow of the system, yet they were predicted to be benign which is correct. The presence of very few false positives was observed. The rule-based detection assigned a low severity rating to these false positives due to their infrequent occurrence.

To validate the IDS system against the DDoS attack, another adversarial scenario was developed. We replicated the scenario of one type of attack where the attacker's goal was to overwhelm the SOFIE platform. The analysis of the network traffic reveals a sudden surge, indicative of anomalous occurrences. The outcomes of the detection are visually presented in Fig.4(b). The proposed IDS solution has successfully identified the IP address associated with malicious flows. Furthermore, the proposed solution provides the frequency of these predicted flows obtained through rule-based detection. The severity of the attack is subsequently determined based on the calculated frequency and classified as low, medium, and high. It demonstrates, for instance, that the DDoS attack characterized in the 3rd row of the table as presented in Fig.4(b) exhibits low severity showing that its frequency of only 1, whereas the preceding two rows is indicative of actual DDoS attacks due to their frequencies surpassing a predetermined threshold. This experimentation utilized the trained Random Forest model.



(a) Dashboard showing the statistics during Normal operation



(b) Dashboard showing the statistics under Attack operation

FIGURE 4. Proposed IDS behavior under Normal and Attack Scenario.

The performance of the IDS system during normal and adversarial scenarios is summarized in Table 3.

We supposed that during the Normal activity, there was no malicious activity and the same assumption was made during the simulation of the malicious activity. The results show that the IDS system has achieved 97.87% Detection Rate (DR) which means that the system has correctly identified 97.87% of normal flows. In a similar way to the malicious flows, the system identified 98.20% of malicious entries. The higher DR for both the normal and malicious activities shows that the system is quite effective in real-time which is crucial for cyber-security applications. An extensive discussion regarding the results of the training and testing phases is provided in the subsequent section.

V. RESULTS EVALUATION

We make use of the different ML algorithms which were trained on the real-time data collected from the supply chain. We make use of the different supervised ML approaches, namely 1. Random Forest (Ranger), 2. Extreme Gradient Boosting (xgboost), 3. Neural Network, 4. k-Nearest Neighbors (kknn), 5. Naive Bayes, 6. Multinomial Logistic Regression, and 7. Logistic Regression. For the unsupervised ML approaches, we make use of the One-Class Support Vector Machine (OC-SVM) which needs only the benign traffic for the training.

These algorithms were trained and later tested in the real-time environment with the new data which was never used in training. These models were evaluated through various ML metrics along with the concept of overall weighted accuracy to test the complete flow of the architecture.

TABLE 4. Description of the confusion matrix.

True Label	Predicted Labels	
	Benign	Malicious
Benign	True Negative (TN)	False Positive (FP)
Malicious	False Negative (FN)	True Positive (TP)

A. PERFORMANCE EVALUATION KEY PERFORMANCE INDICATORS

In the ML domain, the trained models can be evaluated through various standard matrices which can be derived through the confusion matrix. The confusion matrix consists

of four states, two states being used to represent the correct predicted decision taken by the algorithm and two states representing the false predicted decisions. The confusion matrix for the flow classifications is shown in Table 4.

The TP is a decision that was originally an attacking flow, and the model has correctly predicted this flow as malicious while the TN is a decision that was originally benign, and the model predicted it as benign. FP is the false prediction which was originally a benign entry while FN is the false prediction which was originally a malicious flow. Through this confusion matrix, we can devise several performance indicators to do the numerical validations of the IDS.

True Positive Rate (TPR) or Detection Rate (DR): It is defined as the ability of the model to correctly identify the positive instance as in (1).

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1}$$

False Alarm Rate (FAR) or False Positive Rate (FPR): It is defined as the measure of the proportions of the negative class that were identified as positive as in (2).

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{2}$$

Along with the above-mentioned performance indicators we also decided to make use of the most widely standard parameters such as:

Accuracy: Accuracy measures the overall correctness of the model, indicating the proportion of correctly predicted instances out of the total instances as in (3)

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{3}$$

Recall: It is a ratio of the true positive predictions done by model with the the total number of actual positive samples as in (4).

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

Precision: Precision is calculated as a proportion of the positive predictions made by the model. It determines the quality of the model as in (5).

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5}$$

F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, with higher values indicating better overall performance as in (6).

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{6}$$

The above-mentioned matrices will help to get a comprehensive understanding of the classification task done by the ML models. These matrices for each of the models are discussed in the below section.

B. PERFORMANCE EVALUATION RESULTS

We evaluated the models through the above-mentioned indicators. First of all, we trained all the ML models using the same evaluation settings. Cross-validation with the 3 folds resampling was used to get the more robust model performance during the training phase and to get the model performance optimized. The features selection was done through wrapper methods and the selected features were used for the training of the models. The initial evaluation of the model was done through the overall accuracy as shown in Fig.5. The best-performing models were Ranger Forest with the highest accuracy of 99.96%, XGBoost with 99.33%, and KKNN with 99.96%. The Naive Bayes score is the lowest accuracy as compared to other algorithms. For further analysis, we make use of several supervised ML key performance indicators. The confusion matrix was generated which helps get some interesting insights about the trained supervised ML models. The results of all these indicators are summarized in Table 5.

Most of the time, the ML-based IDS are evaluated on the TPR and FPR. We observed that supervised ML algorithm performance was quite impressive for almost all the selected algorithms. The Ranger Forest, which is a fast implementation of the random forest has achieved an overall highest accuracy and highest TPR as compared to other algorithms and the lowest FPR as compared to others. From the table we can observe that the best algorithm selection cannot be done based on a single performance parameter as the performance of all the algorithms with the training and testing data is very slightly different from each other. During the training phase of these models, we calculated the ranking of these models using cross-validation based on the AUC (Area Under the Curve) both for the training and testing set. Lower mean ranks indicate better performance, as the ranks are computed in descending order of AUC values and Ranger Forest has achieved the lowest AUC value. As the trained model obtained from the training stage must be deployed where new data obtained from the real-time setting is used to validate the system's performance. In industrial environments, the IDS system's goal is to detect those Normal activities correctly from the malicious traffic. The training dataset contains imbalanced data containing larger number of records for the normal entries while having a smaller attack traffic. The IDS system's objective was to perform the detection in real-time where false detection can cause serious damages. The choice of the optimal criteria of selecting a model focused on having larger TPR and less FPR. Based on this selection criteria, Ranger Forest was selected and used in the deployment infrastructure to perform the predictions.

For the unsupervised ML approach, we utilized the OC-SVM algorithm which is an extended version of the SVM that can be trained by using the benign data obtained from the infrastructure. The algorithm tries to learn the normal behavior of the benign data for the anomaly detection task. The algorithm performance depends on the optimization parameters which must be selected appropriately to maximize

TABLE 5. Performance of supervised machine learning evaluated on validation data.

Sr. No	Algorithm	Accuracy	Precision	Recall	F1 Score	TPR	FPR	Rank
1	Naive Bayes	0.9528	0.9006	0.9988	0.9471	0.9988	0.0811	7
2	Ranger Forest	0.9997	0.9998	0.9996	0.9997	**0.9996**	0.0001	1
3	Log Regression	0.9991	0.9994	0.9988	0.9990	0.9988	0.0005	5
4	KKNN	0.9996	0.9997	0.9996	0.9996	**0.9996**	0.0002	2
5	xgboost	0.9993	0.9995	0.9990	0.9992	0.9990	0.0003	3
6	Multinom	0.9990	0.9992	0.9988	0.9990	0.9988	0.0007	4
7	Neural Network	0.9596	0.9996	0.9212518	0.9588281	0.9212518	0.0003836279	6

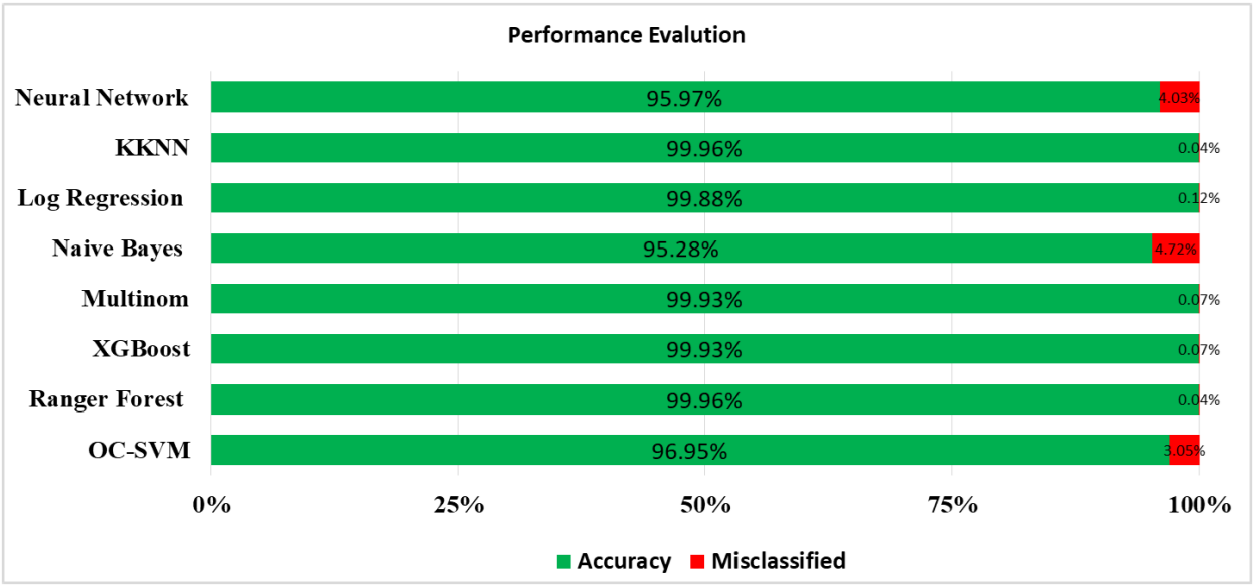


FIGURE 5. Different models accuracy comparisons.

the system performance. The normal flows from the training data were only used to train the algorithm. The performance of the OC-SVM on different distributions of the data is shown in Table 6. For the 30% training data, the model achieves reasonable performance suggesting that the smaller amount of the data it can effectively learn the patterns and result in achieving a higher accuracy of 98.6%. With the 50% training data, the model performance improves both in terms of accuracy and FPR reduced. It seems that the model is overfitting thus resulting in 100% FPR. For 80% of the training data, tuning the hyper-parameters gets complicated and this results in increased FPR. It is worth mentioning that all the supervised algorithms achieve less FPR rate as compared to the OC-SVM algorithm. Thus, only supervised models were used for the validation purpose in a real-time manner.

We selected the top models based on the above-mentioned selection criteria which were tested and deployed into the real-time environment. The normal flows of the system obtained from the normal operations of the system and the simulation of the attacks in the controlled environment were used to evaluate the performance of the proposed solution. The summary of their performance is shown in Fig. 6.

TABLE 6. Performance of the un-supervised OC-SVM on different distribution of data.

Algorithm	Accuracy	Precision	Recall	F1 Score	TPR	FPR
30-OC-SVM	0.986	0.985	0.999	0.992	0.999	0.044
50-OC-SVM	0.989	0.987	1	0.993	1	0.041
80-OC-SVM	0.975	0.985	0.983	0.984	0.983	0.048

Overall, all the algorithms achieved an overall weighted accuracy greater than 96%. in a controlled environment. The ML models have done the prediction on each flow level which is then employed by the rule-based detection. The rule-based detection has further summarized these results and different severity (High, Medium, Low) were assigned based on their frequency. These results help in taking control action by the network administrator. For the DDoS attack, the suitable action is to block the malicious entity's ip address.

VI. DISCUSSION

One common problem with the existing research efforts is that there is no clear discussion about how the models were trained and how they can be applicable in a real-time scenario. Another common issue was that the accuracy results

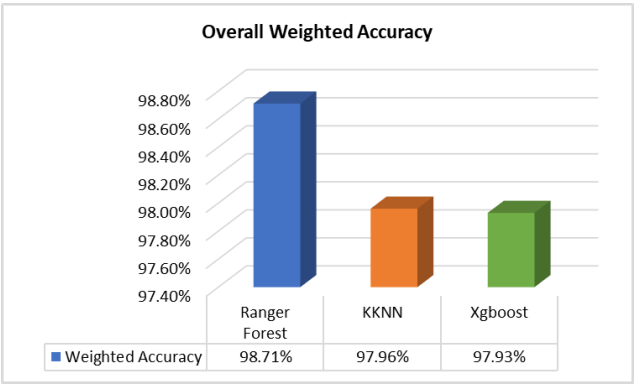


FIGURE 6. Performance of supervised models in real-time scenario.

were very high which can be an indication that the training process has suffered the model overfitting. It can happen when the model hyperparameters are not properly optimized. This problem was considered while preparing the model. Khan et al. [8] has proposed a framework to handle the issue associated with the supply chain system. A direct comparison cannot be due to differences in the training dataset. The system makes uses of federated learning and achieve an overall accuracy of 99.33% which is very high. Similarly, Huma et al. [34] also achieved an accuracy ranging between 97.6% to 100%. The common problem with this system was that there is no discussion on the usability and validation with real-time industrial systems.

Uszko et al. [9] has presented a similar methodology of using rule-based detection along with ML-based detection. They achieved an efficiency of the system 98.57%. The rule-based detection was used to achieve the early detection while the ML was used to detect the sophisticated attack. In this work, we first employed the machine learning model to detect the attack and then followed by the rule-based detection. The IDS system was trained and validated in the real-time industry of farm to fork thus results in achieving a training accuracy of 99.96% and validation accuracy of 97.87%. To ensure that the model does not suffer overfitting, the k-fold cross-validation and data normalization were used to make sure that the training process was smooth. The validation results show that the proposed system works better in the real-time scenario while there was no validation in real-time done by the existing approaches.

The detection results of machine learning and rule-based detection are integrated to avoid triggering alerts for each malicious flow. Initially, the trained ML model performs prediction at each flow level. It results in generating a high volume of alerts if these predictions are used directly to trigger the alerts. The rule-based detection further enhances these prediction results and calculates the frequency of each network flow based on aggregation using the timestamp. If only ML-based detection is used, then it may result in generating many alerts during a DDoS attack and it will be hectic for the network administrator to take some actions against the malicious entities individually as the attacker can use multiple IP addresses or bots to launch this attack. If only rule-based detection is used, then the attacker can use sophisticated techniques to bypass these rules easily. To take advantage of both approaches, we combined them to achieve a low false positive and aggregated results.

The attack detection flow consists of ML-detection results followed by rule-based detection. The primary reason for employing rule-based detection following machine learning detection is to provide an additional layer of security and enhance the detection capability of the overall system which ultimately can be used to take the mitigation actions. The proposed work is a proof of study that makes use of machine learning and rule-based detection to work in real-time industry scenarios and was validated with one particular category of DDoS attack. As the real-time traffic is being used which contains individual bidirectional network flows, the model tries to perform the prediction at each flow level. The individual flows are classified as normal or abnormal. After employing the rule-based detection these flows are further summarized resulting in providing useful insight into the attacking entries. This information is useful in real-time use cases to take immediate mitigation actions. If only the machine learning results are used to take the mitigation actions, then they will result in an increased number of alerts generation for every malicious network flow which is not convenient for the network administrator to take some actions. The rule-based detection simplifies this alert generation mechanism by utilizing the domain-specific knowledge or expert rules designed for each particular type of attack which allows the security network administrator to tailor responses that align with the kind of threat, e.g. in DDoS blocking the malicious IP address.

VII. CONCLUSION AND FUTURE DIRECTION

The existing IDS solutions lack in providing appropriate levels of protection against attacks targeting critical infrastructures. One major issue they faced was that they utilized a combination of real-time and public datasets to train and validate their models. Moreover, they lack real-time evaluation of these systems in practical scenarios. In this paper, we propose an efficient IDS solution to protect CPPS against DDoS attacks. The proposed IDS solution makes use of the real-time network traffic obtained from an industrial scenario, particularly a so-called F2F supply chain system. The proposed solution has utilized the seven ML approaches to achieve better attack detection capability and rule-based detection to complement the results generated by the ML models. The real-time benign and attack network traffic was collected in a controlled environment by making use of the tcpdump. The CIC-FLOW METER was used to extract network traffic features. The wrapper methods were used to extract the most influencing features that were used for the training of the ML models. The IDS solution employed several supervised and un-supervised ML approaches for anomaly detection in the network flows. Ranger Forest, KKNN, and xgboost outperform betters in detecting the

DDoS attacks with an Accuracy > 99.93% and TPR > 99.90%. For unsupervised approaches, different distributions of the data were used to train the OC-SVM algorithm. It achieved an Accuracy > 98.9% and TPR > 98.3 %.

The experimental results have shown that supervised ML algorithms perform better as compared to unsupervised. The ML models perform predictions for each network flow. We utilized rule-based detection to calculate the frequency of predicted malicious flows over time. A high frequency indicates the possibility of DDoS attack and assigns different severity levels to the predicted flows. The proposed IDS solution with top-performing supervised ML models with rule-based detection was evaluated further with real-time simulation of attack and normal traffic obtained at different intervals of time. It correctly identifies the benign and malicious network entities and achieves an overall weighted accuracy of 98.71%. The proposed IDS solution has achieved an overall high accuracy in the training, testing, and validation phases.

For future work, this study can be utilized to check efficiency against other types of attacks that can impact the CPPS. The performance evaluation of the ML model along with ensemble learning approaches and rule-based detection for other types of attacks needs to be investigated. Developing a collaborative IDS that makes use of different types of techniques such as rule-based detection, behavior modeling, and ML-based approaches to predict the different attack paths can be explored in the future. The proposed IDS solution should consider alert aggregation for different outputs to see the correlation between alerts generated by rule-based detection and ML models which can be used for predicting the future attack stage. Other attack scenarios such as stealthy attacks, and data integrity attacks can cause serious damage to the supply chain system. The proposed IDS system will be extended to work in a more generic so that its scalability can be done in other environments.

REFERENCES

[1] G. Karpagam, B. V. Kumar, J. U. Maheswari, and X.-Z. Gao, *Smart Cyber Physical Systems*. New York, NY, USA: CRC Press, 2020.

[2] H. C. Verma, S. Srivastava, T. Ahmed, and N. A. Usmani, "Cyber threats in agriculture and the food industry: An Indian perspective," in *Advances in Cyberology and the Advent of the Next-Gen Information Revolution*. Hershey, PA, USA: IGI Global, 2023, pp. 109–122.

[3] X. Koufteros and G. Lu, "Food supply chain safety and security: A concern of global importance," *J. Marketing Channels*, vol. 24, nos. 3–4, pp. 111–114, 2017.

[4] A. Yeboah-Ofori, S. Islam, S. W. Lee, Z. U. Shamszaman, K. Muhammad, M. Altaf, and M. S. Al-Rakhami, "Cyber threat predictive analytics for improving cyber supply chain security," *IEEE Access*, vol. 9, pp. 94318–94337, 2021.

[5] D. E. Comer, *Computer Networks and Internets*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[6] J. David and C. Thomas, "DDoS attack detection using fast entropy approach on flow- based network traffic," *Proc. Comput. Sci.*, vol. 50, pp. 30–36, Jan. 2015.

[7] F. B. Saghezchi, G. Mantas, M. A. Violas, A. M. de Oliveira Duarte, and J. Rodriguez, "Machine learning for DDoS attack detection in industry 4.0 CPPSs," *Electronics*, vol. 11, no. 4, p. 602, Feb. 2022.

[8] I. A. Khan, N. Moustafa, D. Pi, Y. Hussain, and N. A. Khan, "DFF-SC4N: A deep federated defence framework for protecting supply chain 4.0 networks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 3, pp. 3300–3309, Mar. 2023.

[9] K. Uszko, M. Kasprzyk, M. Natkaniec, and P. Chołda, "Rule-based system with machine learning support for detecting anomalies in 5G WLANs," *Electronics*, vol. 12, no. 11, p. 2355, May 2023.

[10] R. R. R. Barbosa and A. Pras, "Intrusion detection in SCADA networks," in *Proc. 4th Int. Conf. Auton. Infrastruct., Manag. Secur. (AIMS)*, Zurich, Switzerland, Jun. 2010, pp. 163–166, doi: 10.1007/978-3-642-13986-4_23.

[11] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of IP flow-based intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 343–356, 3rd Quart., 2010.

[12] P. Borges, B. Sousa, L. Ferreira, F. B. Saghezchi, G. Mantas, J. Ribeiro, J. Rodriguez, L. Cordeiro, and P. Simoes, "Towards a hybrid intrusion detection system for Android-based PPDR terminals," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 1034–1039.

[13] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, pp. 18–28, Feb./Mar. 2009.

[14] S. S. Abosuliman, "Deep learning techniques for securing cyber-physical systems in supply chain 4.0," *Comput. Electr. Eng.*, vol. 107, Apr. 2023, Art. no. 108637.

[15] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 1827–1834.

[16] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, pp. 621–636, Mar. 2018.

[17] *A realistic cyber defense dataset (CSE-CIC-IDS2018)*. Accessed: Apr. 8, 2023. [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018

[18] 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. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.

[19] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[20] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[21] E. Chatzoglou, G. Kambourakis, and C. Kolias, "Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset," *IEEE Access*, vol. 9, pp. 34188–34205, 2021.

[22] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023.

[23] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," *Sci. Afr.*, vol. 9, Sep. 2020, Art. no. e00497.

[24] C. Ioannou and V. Vassiliou, "Classifying security attacks in IoT networks using supervised learning," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 652–658.

[25] A. S. Ahanger, S. M. Khan, and F. Masoodi, "An effective intrusion detection system using supervised machine learning techniques," in *Proc. 5th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Apr. 2021, pp. 1639–1644.

[26] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Comput. Commun.*, vol. 35, pp. 772–783, Apr. 2012.

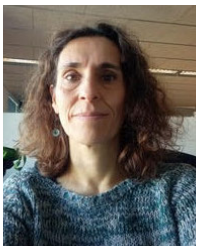
[27] A. Hussain, F. Aguiló-Gost, E. Simó-Mezquita, E. Marín-Tordera, and X. Massip, "An NIDS for known and zero-day anomalies," in *Proc. 19th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Apr. 2023, pp. 1–7.

[28] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustain. Cities Soc.*, vol. 72, Sep. 2021, Art. no. 102994.

[29] A. Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, pp. 83965–83973, 2020.

[30] A. Alzahrani and T. H. H. Aldhyani, "Design of efficient based artificial intelligence approaches for sustainable of cyber security in smart industrial control system," *Sustainability*, vol. 15, no. 10, p. 8076, May 2023.
[31] A. L. Perales Gomez, L. F. Maimo, F. J. G. Clemente, J. A. M. Morales, A. H. Celdran, and G. Bovet, "A methodology for evaluating the robustness of anomaly detectors to adversarial attacks in industrial scenarios," *IEEE Access*, vol. 10, pp. 124582–124594, 2022.
[32] W. Wang, P. Yi, J. Jiang, P. Zhang, and X. Chen, "Transformer-based framework for alert aggregation and attack prediction in a multi-stage attack," *Comput. Secur.*, vol. 136, Jan. 2024, Art. no. 103533.
[33] M. F. Elrawy, L. Hadjidemetriou, C. Laoudias, and M. K. Michael, "Detecting and classifying man-in-the-middle attacks in the private area network of smart grids," *Sustain. Energy, Grids Netw.*, vol. 36, Dec. 2023, Art. no. 101167.
[34] Z. E. Huma, S. Latif, J. Ahmad, Z. Idrees, A. Ibrar, Z. Zou, F. Alqahtani, and F. Baothman, "A hybrid deep random neural network for cyberattack detection in the industrial Internet of Things," *IEEE Access*, vol. 9, pp. 55595–55605, 2021.
[35] M. O. Pahl and F.-X. Aubet. (2018). *DS2OS Traffic Traces IoT Traffic Traces Gathered in a the DS2OS IoT Environment*. [Online]. Available: https://www.kaggle.com/francoisxa/ds2ostraffictraces
[36] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky, "Attack graph model for cyber-physical power systems using hybrid deep learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4007–4020, Sep. 2023.
[37] Y. Hu, H. Li, T. H. Luan, A. Yang, L. Sun, Z. Wang, and R. Wang, "Detecting stealthy attacks on industrial control systems using a permutation entropy-based method," *Future Gener. Comput. Syst.*, vol. 108, pp. 1230–1240, Jul. 2020.
[38] H. C. Leligou, A. Lakka, P. A. Karkazis, J. P. Costa, E. M. Tordera, H. M. D. Santos, and A. A. Romero, "Cybersecurity in supply chain systems: The farm-to-fork use case," *Electronics*, vol. 13, no. 1, p. 215, Jan. 2024.
[39] J. Giraldo, D. Urbina, A. Cardenas, and J. Valente, "A survey of physics-based attack detection in cyber-physical systems," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Jul. 2018.
[40] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 4, no. 1, pp. 48–59, Mar. 2018.
[41] V. S. Dolk, P. Tesi, C. De Persis, and W. P. M. H. Heemels, "Event-triggered control systems under denial-of-service attacks," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 1, pp. 93–105, Mar. 2017.
[42] A. Amodei, D. Capriglione, L. Ferrigno, G. Miele, G. Tomasso, and G. Cerro, "A measurement method for intrusion detection in cyber IoT data stealing attacks," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2023, pp. 1–6.
[43] L. A. Maglaras and J. Jiang, "Intrusion detection in SCADA systems using machine learning techniques," in *Proc. Sci. Inf. Conf.*, Aug. 2014, pp. 626–631.
[44] B. Cetin, A. Lazar, J. Kim, A. Sim, and K. Wu, "Federated wireless network intrusion detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 6004–6006.
[45] A. De Montigny-Leboeuf, M. Couture, and F. Massicotte, "Traffic behaviour characterization using NetMate," in *Recent Advances in Intrusion Detection* (Lecture Notes in Computer Science), vol. 5758, E. Kirda, S. Jha, and D. Balzarotti, Eds., Berlin, Germany: Springer, 2009, pp. 367–368, doi: 10.1007/978-3-642-04342-0_27.
[46] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related," in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2016, pp. 407–414.
[47] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *ICISSp*, vol. 1, 2018, pp. 108–116.
[48] Y. Wang, G. Xiong, Z. Li, M. Cui, G. Gou, and C. Hou, "WSNet: A wrapper-based stacking network for multi-scenes classification of DApps," in *Web and Big Data* (Lecture Notes in Computer Science), vol. 13421, B. Li, L. Yue, C. Tao, X. Han, D. Calvanese, and T. Amagasa, Eds., Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-25158-0_13.
[49] (May 16, 2023). *F2F Use Case Demo*. Accessed: Aug. 4, 2024. [Online]. Available: https://fishy-project.eu/blog/f2f-use-case-demo

AYAZ HUSSAIN received the M.S. degree in computer engineering from the University of Engineering and Technology at Taxila (UET Taxila), Pakistan, in 2018. Currently, he is pursuing the Ph.D. degree in cybersecurity with Universitat Politècnica de Catalunya, Barcelona Tech-UPC. His Ph.D. thesis focused on developing predictive analytics-based strategies to protect critical systems. He is currently working on developing AI-based strategies to protect the system against cyber attacks. He has actively contributed to EU-funded projects. His potential research interests include working in the domain of artificial intelligence, the IoT, cloud computing, cybersecurity, and industrial systems.



EVA MARÍN TORDERA received the M.S. degrees in physics and electronic engineering from the University of Barcelona, in 1993 and 1998, respectively, and the Ph.D. degree from the Technical University of Catalonia (UPC), in 2007. She has been an Associate Professor with UPC, since 1996, where she has developed her teaching and research activities in the Computer Architecture Department. Her research started at the Advanced Broadband Communications Center, in 2000, where she introduced a new prediction-based paradigm for routing in optical networks. In 2008, she moved her activities to the Advanced Network Architectures Laboratory (CRAAX), where she is actively working on transferring network concepts to society (the IoT, cloud, and cybersecurity).

XAVI MASIP-BRUIN received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Technical University of Catalonia. He is currently a Full Professor with UPC, and the Director of the Advanced Network Architectures Laboratory (CRAAX). His publications include more than 200 papers in international refereed journals and conferences. He has participated and/or led many national and regional projects and EU contracts and has also led contracts with the industry. In early 2013, he co-founded MATPOL Technologies, a Bay area based start-up. He has chaired and co-chaired many international conferences, including ONDM, WWIC, NOC, Saconet, Med-Hoc-Net, IEEE IWQoS, and DRCN. His contributions were recognized with a 2016 IBM Faculty Award. He serves as an Editor for *Optical Switching and Networking* (OSN) and *Computer Communications* journals.

HELEN C. LELIGOU (NELLY) received the Dipl.-Ing. and Ph.D. degrees from the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece. From 2007 to 2017, she was an Assistant Professor with the Technological Educational Institute of Sterea Ellada. She is currently an Associate Professor with the University of West Attica. She is also involved in blockchain technologies and their combination with artificial intelligence and federated learning techniques. She is/was a scientific coordinator of the LIFE-GENERA and H2020-ASSET Project. She has participated in more than 20 EU-funded ACTS, 1ST, ICT, and H2020 research projects in the above areas and also acts as an evaluator for national and EU funded proposals. Her research interests include computer networks and information-and-communication-technologies, such as security mechanisms, routing protocols and trust management in wireless sensor networks, control plane technologies in broadband networks, including HFC, PON, WDM metro, and core networks, embedded and network system design and development, and the IoT enabled solutions for different application sectors like energy efficiency/optimization in buildings and affect detection in learning environments.