Machine Learning-based Industrial LAN Networks Using Honeypots

Abbasgholi Pashaei*[1], Mina Zolfy Lighvan[2], Asghar Charmin[3]
1,3- Department of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran
2- Department of Electrical and Computer Engineering Faculty, Tabriz University, Tabriz, Iran
Email: a-pashaei@iau-ahar.ac.ir(Corresponding author), mzolfy@tabrizu.ac.ir , a_charmin@sut.ac.ir
Receive Date: 10 September 2022, Revise Date: 6 October 2022, Accept Date:20 November 2022

Abstract
The emergence of industrial Cyberinfrastructures, the development of information communication technology in industrial fields, and the remote accessibility of automated Industrial Control Systems (ICS) lead to various cyberattacks on industrial networks and Supervisory Control and Data Acquisition (SCADA) networks. Thus, it is essential to continuously improve the security of the networks of industrial control facilities. The purpose of honeypots is to deceive the attackers so that we may learn about their tactics and behavior. Security professionals gather all pertinent data on attack methods and behavior and take decisive action to tighten security controls. The simulation results demonstrate the ML-based mechanism's efficiency in monitoring the ICS panel for detection approaches. Therefore, the designed system for early intrusion detection can protect industrial systems against vulnerabilities by alerting the shortest possible time using online data mining in the EIDS database.

Keywords: Intrusion Detection System, Honeypot, Network Security, Machine Learning.

1. Introduction

Nowadays, cyberattacks on Industrial Control Systems (ICSs) are on the rise. As a result, Industrial Smart Sensors (ISSs) and users' information are vulnerable to these attacks [1]. For this purpose, information security and monitoring of Industrial Intelligent Sensors (IISs) have been considerably expanded with the advent of Information and Communication Technology (ICT) in the present age [2]. Cybersecurity information technology can upgrade communication and network security by providing efficient and practical approaches. Therefore, advanced techniques have been needed to increase the network security level and intrusion detection [3]. Furthermore, with the development of technology, cyberattacks against the information of network users have also been changed. Therefore, security systems must constantly be updated to deal with any zero-day attack [4].

The SCADA framework moves forward the controlling and administering mechanical systems by synchronizing the machine's distinctive parts using the amounts assembled by sensors [5]. Assurance against cyberattacks for the SCADA could be a strict requirement, but it is an essential tool to preserve network security. Using a Honeypot detection system that has the ability to maintain high security can be advantageous in complex industrial systems with high adaptability [6]. In this manner, Honeypots are used to ensure ICS facilities from being assaulted and explore assailants' behavior unauthorized as well as identify to obscure assaults [7].

In this investigation strategy, Honeypots are utilized as a frontline detector to distinguish vulnerabilities and decrease assaults. Various algorithms to classify ML-based techniques are also used for early detection of infiltration with Honeypot [8]. These classification algorithms include Support Vector Machine (SVM), Decision Tree (DT) [9], Multi-Layer Perceptron

(MLP) [10], Dense Layer [11], K-Nearest Neighbor's (KNN) [12], 1D Convolutional Neural Network (CNN1D) [13], and Long Short-Term Memory (LSTM) [14]. In this research, the architectural design of early IDS with ML-based Honeypot is proposed as a solution for the detection of intrusion. The proposed method is presented in this research.

In this paper, a general Honeypot industrial IDS are reviewed in 1,2. The framework and proposed methodology is reviewed in 3, also configuration and assembling honeypot using ML systems is studied in sections 4. The Honeypot performance in ICS protection using the ML system in section 5. The discussion on the EIDS in sections 6. Next, the EIDS technique is discussed and summarized as a

n ICS application in section 7, and Finally, the conclusion is examined in section 8 individually.

2. Related Works

Security is a significant challenge to satisfy the requirements of SCADA applications in ICS. A broad range of approaches has been presented in the literature to address this issue. The authors in [15] presented a novel idea with the Honeypot system and IDS. The primary purpose in [15] is to mislead the attackers, get their information, and lead them in the wrong direction. In [16], the design of an intelligent grid Honeypot system has been proposed. Comprehensive information on Honeypot-based techniques used in smart grids had been reviewed in [17]. Dynamic property Honeypot based on Blockchain has been studied in [18], which distinguishes between real and fake resources in the system. To modify IoT security, Pashaei et al. [19] described an early state intrusion detection strategy using honeypots in an industrial network environment. The idea behind this work was to use the Markov process and action-reward policies to continuously observe network events. The state-based honeypot event analysis model was set up to detect and isolate distributed DoS attacks and Man-in-the-Middle attacks. As previously stated, this work used reinforcement-learning principles and Markov chain principles to detect suspicious events as quickly as possible. Furthermore, this scheme guaranteed its higher accuracy rate based on the best reward weights based on changing event states. In [20], a Honeypot system is proposed for network intruders, which evaluate network server security techniques using the Honeypot system. The combination of Honeypot with IDSs has been used to increase effective detection in [21]. This novel approach developed a new Honeypot technique-based security approach for real-time intrusion detection and prevention systems. In [22], a multi-faceted and multi-phased method is introduced to design an IoT Honeypot ecosystem. This approach applies to obtain information from cyberattackers and examine them in the IoT systems using Honeypots.

In [23], an IDS for SCADA-based systems exploiting DL networks has been proposed to protect ICSs from conventional and network-based cyberattacks. It uses a convolutional neural network for salient temporal traffic pattern modeling and uncertain time window identification instead of using hand-crafted features from individual flows. Besides, it has introduced a re-training mechanism to handle new attacks, and analysis results on realistic SCADA traffic datasets indicate that a DL-based system improves cyberattack detection

accuracy and identifies advanced-emerged threats. In [24], another DL-based IDS has been presented to detect temporally correlated and uncorrelated attacks in SCADA-based systems. This method combines a feedforward neural network and LSTM to enhance the IDS performance in temporal correlation detection among the data flow.

To detect the attacks that do not already occur in databases, [25] has also exploited ML algorithms' capabilities. In this research, the authors have employed a real dataset aggregated from a gas pipeline system from the Mississippi State University to present an IDS in SCADA networks. The presented system evaluates four methods to estimate missing data and two mechanisms to normalize data. Then, it analyzes the accuracy and precision of the SVM and RF in IDSs. Experimental results demonstrate that the RF algorithm could detect intrusions efficiently, and then it is a reasonable method to guarantee the security of SCADA networks in ICS applications. However, ML-based approaches focus on cyberattack detection against SCADA networks in industrial environments, and they do not describe the real impact of threats on ICS.

In [26], a cyber-physical identification plan has been introduced for evaluating risk levels of intrusions against the industrial systems that suffer from vulnerabilities of control devices and protocols to address the challenges mentioned above. The method extracts communication patterns and states of devices to characterize the system structure. Any violation of the plan is also identified as a false or network-based cyberattack. It also presents a risk assessment mechanism to estimate the impact factor of each attack on ICS by combining SCADA status and network intrusions. Performance evaluation of the plan shows that providing relevant information about network administrators' threats improves the analyzing process of cybersecurity attacks against the SCADA networks.

In [27], a hybrid multi-level anomaly prediction method for intrusion detection in SCADA networks to deal with unbalanced datasets in ICS applications has been presented. It exploits the anticipated nature of communication patterns in a gas pipeline SCADA network to enhance cyberattack detection accuracy. The mechanism preprocesses data to standardize them, uses a dimensionality reduction method to improve the anomaly detection process's performance, employs an edited KNN rule plan to scale the dataset, and creates a Bloom filter-based signature database for some time without abnormality occurrence. Furthermore, it combines the contents-level detection method with an instance-based learning approach to improve the anomaly detection process's accuracy. Although the technique focuses on the cybersecurity arena industrial, it ignores the process states in industrial applications' physical environments. In [28], a secure mechanism for detecting cyber and physical aggression in SCADA networks is introduced to tackle physical field challenges and detect processing attacks such as the Man-in-the-Middle (MITM) attack. It uses the validation of process states to identify users' malicious behaviors and prevent the physical equipment from damages caused by processing attacks. The mechanism has also proposed a Nonparallel Hyperplane-based Fuzzy (NHF) classifier for dataset classification. The

comparisons prove this hybrid mechanism's performance is preferable to the parallel hyperplane of the SVM in the cyber field.

In [29], a reliable host-based IDS through the OS diversity to detect new kinds of threats in SCADA networks has been introduced. SCADA communications over time are evaluated in an ICS to select the most reliable OS in this system. Experiments show that choosing the most suitable OS enhances IDS accuracy compared to the single operational system-based environments. Physical and network metrics with IDS to support SCADA networks' security have been combined in [30]. The approach is simulated on a gas pipeline dataset to indicate the DT's efficiency in classifying various categories of features. Although the combined IDS-based methods improve effectiveness in SCADA networks in ICS, they struggle with unacceptable performance.

Based on the related researches mentioned above, to resolve challenges in false-positive IDS, a new design for detecting attacks becomes necessary. As a result, it is more efficient to develop a new IDS for protecting ICS from cybersecurity attacks to guarantee critical applications' requirements.

3. Proposed Methodology

ICSs are a combination of different industrial equipment types, which perform twenty-four hours seven days (24*7) a week in a stable and operational state and support continuous, steady-state, and full-time processes. ICSs include metering equipment with precise calibration for measuring analog and digital equipment, and intelligent data collection devices, in which collected data are transmitted securely through communication and telecommunication equipment. Measuring sensors compatible with the telemetry function installed in equipment located in industrial facilities help with this. ICSs also include all hardware types of equipment, software programs, and various communication equipment to convert received information from sensors into readable information for the operation centers and monitor industrial facilities.

As these control devices are specially used in industry, they offer a new large-scale industrial private system with superior performance that can handle complex control and processing work in industrial facilities. Connecting industrial and intelligent automation is an essential and valuable asset for industrial enterprises such as petrochemicals and oil and gas refineries, which enables them to perform in hazardous control environments such as oil and gas separators facilities that no system can work in. In contrast, traditional industrial control automation is not able to work in these hazardous situations. Since the industrial devices were equipped with a Local Area Network (LAN) and wireless, the number of cyberattacks has increased. However, most industrial control equipment used in critical infrastructure is proprietary, and computer equipment uses proprietary communication protocols that are not connected to the world outside the local area network and are specially protected.

In order to overcome the mentioned explanations in the previous contents, a combination strategy is utilized for information collection, information investigation, and information revelation in the proposed integrated EIDS architecture, according to Fig. 1. As shown in Fig. 1, seven ML algorithms

models are used to achieve a precise prediction model for analyzing data stored in the EIDS server database. To identify attacks against ICS and record their performance metrics, SVM, MLP, KNN, DT, Dense Layer, CNN1D, and LSTM are used as classification algorithms. To accomplish dimension unification, the article used a well-liked normalizing technique. The normalization equation is as follows:

$$y_i = \frac{x_i - \hat{x}}{\delta}$$
$$\hat{x} = \frac{1}{k}\sum_{i=1}^{k} x_i \quad (1)$$
$$\delta = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}\left(x_i - \hat{x}\right)^2}$$

Where $x_1, x_2, \ldots, x_k$ is a record of raw data. The equation will change each value x_i into a different value y_i, and the resultant data record will be $y_1, y_2, \ldots, y_k$ is made up of all y values. Finally, the new data record's standard deviation is zero and its variance is one. This resolves the issue of big dimension features influencing training results. According to the result of implementing ML algorithms on the world's leading databases, a model is designed to recognize and show anomaly behaviors of data collected from the EIDS network. In general, ML algorithms models includes of basic structure of an autoencoder. An autoencoder must be constructed in three simple steps: an encoder, a decoder, and the setting of a loss function. The majority of encoding stages will translate high-dimension data into low-dimension space through feature reduction. The autoencoder's encoding procedure may be summarised as follows:

$$H = f_\theta(X) = \sigma\left(W_{ij}X + b_{ij}\right) \quad (2)$$

where σ is the activation function and X is the input data.The neuron unit's weight is W_{ij}, and its bias is b. Tanh activation is utilized and calculated in this research by:

$$\text{Tanh}(t) = \frac{1-e^{-2t}}{1+e^{-2t}} \quad (3)$$

For the purpose of network training, the decoding procedure reconstructs the data so that it may be given back to the neural network with the loss function. Equation (4), in which Y are the known class labels and P are the predicted probabilities, illustrates how the network was trained to maximize a logistic loss function:

$$L(Y,P) = \frac{-1}{N}\left(\sum_{i=1}^{N}\sum_{j=1}^{K} Y_{i,k}\log\left(P_{i,k}\right)\right) \quad (4)$$

$Y_{i,k} = \begin{cases} 1 \\ 0 \end{cases}$ where 1, if sample I belongs to class k and 0, otherwise.

Decoding may often be thought of as the opposite of the encoding process. An abbreviated definition of decoding is given as follows:

$$Y = g_\theta(X) = \sigma\left(W_{jk}X + b_{jk}\right) \quad (5)$$

A crucial component of the gradient descent procedure is the loss function. The Mean Squared Error (MSE), which is often used in the autoencoder model, is adopted due to the reconstruction process in the decoding phase. To update the weights and bias of the neuron units, the loss will be transmitted back to the hidden layer. Thus, the following is a presentation of the network's overall loss function:

$$J(W,b)=\frac{1}{2N}\sum_{n=1}^{N}\|Y_n-X_n\|^2 \qquad (6)$$

where N is the overall sample count. This design also enables to obtain of low False-Positive Rates (FPR) to identify and model incoming traffic attacks.

Fig. 1 shows the schematic of the proposed research architecture, including the relationship of industrial indoor systems to servers, firewalls, switches, Wireless Fidelity (WiFi), actuators, sensors, etc., that form the overall architecture of EIDS.

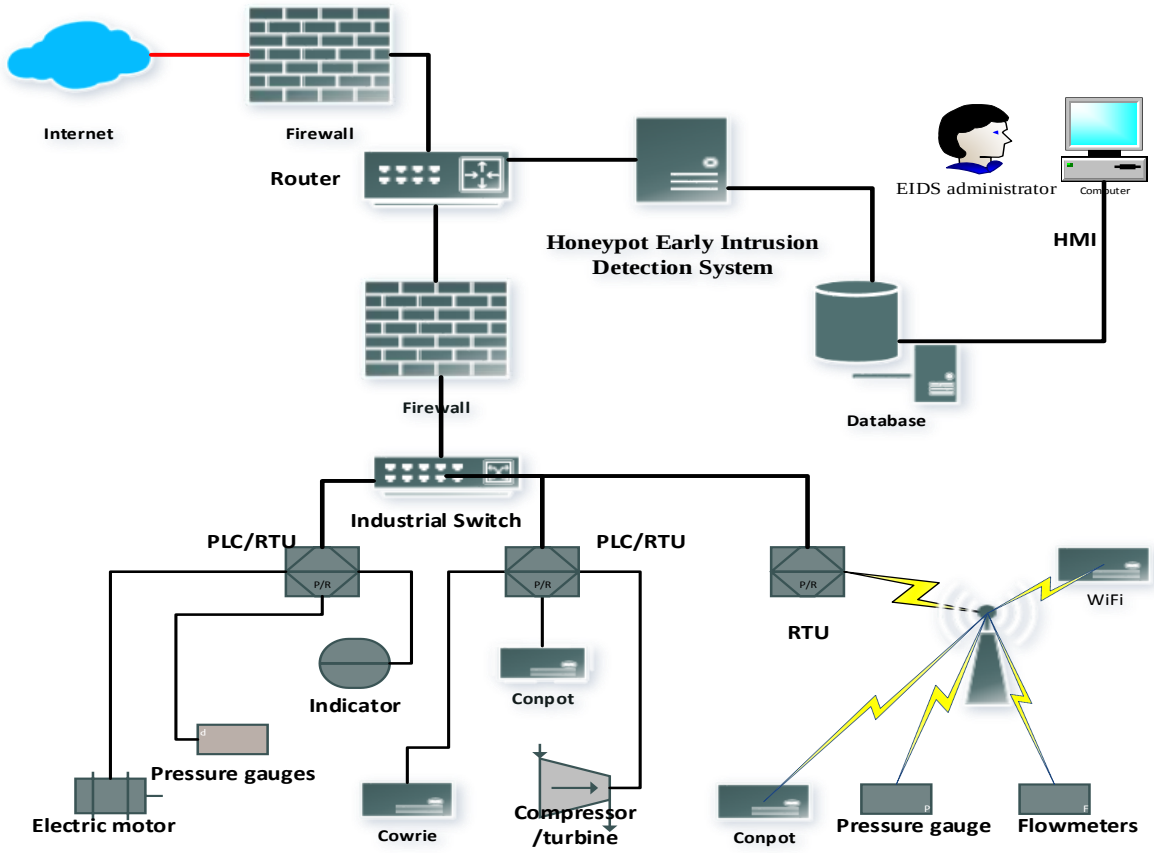


Fig. 1. The overall scheme of proposed integrated EIDS architecture.

4. Configuration and Assembling

The prerequisites for introducing, designing, and programming the EIDS, which incorporates open-source Linux programs and equipment, will be briefly explained and discussed in subsections A to D.

A. Router

The router is the central routing framework for controlling, monitoring, and reading the traffic of industrial and non-industrial networks. Its OS is programmed to store routing in an Internet Protocol table (IPtable) internal memory and communicate well with network

equipment distributed across the facility. Therefore, to control and protect the network against attackers, a copy of the router traffic packets will be sent to the IDS system to check the status of the packets in terms of being normal or abnormal.

To detect attackers analyzing incoming traffic is necessary. For this reason, by activating the Sniffer mode in the Mikrotik router, IDS Snort is allowed and has access to read and receive incoming traffic packages. Therefore, all packets will be read from incoming traffic logs and stored in the database to analyze packets suspected of attacking the attackers.

B. IDS Snort

The anomaly packet discovery is a framework for identifying assaults by proposed EIDS. For this purpose, input traffic from the router to IDS snort is compared to IPtables rules until IDS Snort warns about a suspicious packet. All updated rules about the latest attacks, downloaded directly from the site or manually applied in IDS Snort, compare attackers' attacks and abnormal behavior on the network with this updated data and report any abnormal behavior for the desired traffic, and it is sent to the system administrator for review.

According to the sentences mentioned above, IDS Linux must be run to check network traffic packages. Fig. 2 shows the successful implementation of IDS Snort after the installation and configuration steps.

```
Reload thread started, thread 0x7f50b0592700 (31172)
Decoding Ethernet
Set gid to 1000
Set uid to 1000

        --== Initialization Complete ==--

   ,,_      -*> Snort! <*-
  o"  )~    Version 2.9.14.1 GRE (Build 15003)
   ''''     By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
            Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
            Copyright (C) 1998-2013 Sourcefire, Inc., et al.
            Using libpcap version 1.8.1
            Using PCRE version: 8.32 2012-11-30
            Using ZLIB version: 1.2.7

            Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
            Preprocessor Object: appid  Version 1.1  <Build 5>
            Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
            Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
            Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
            Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
            Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
            Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
            Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
            Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
            Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
            Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
            Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
            Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
            Preprocessor Object: SF_POP  Version 1.0  <Build 1>
            Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
Commencing packet processing (pid=31171)
```

Fig. 2. Sniffing the network by IDS snort.

C. WiFi Honeypot Networks

In industrial environments, Radio communication equipment is used in terms of environmental pollution conditions. Consider that the Radio-frequency (RF) environment uses IEEE 802.11 protocols. Therefore, most

vulnerabilities in the wireless network environment are performed using this protocol in telecommunication cyberspace. An RF device with WiFi router capability was intended for hardware configuration to be suitable for analyzing attacker data. Extensions were used in Honeypot WiFi to maintain essential functions. Depending on the selected device, a range of possible OSs were used, such as Ubuntu 18.04 LTS, Kali 20.4, Conpot. The mentioned equipment and tools should theoretically perform RF monitoring functions, data traffic evaluation scripts, and implement a virtual access point for attackers.

A permanent configuration is created to monitor RF environment cyberspace using Honeypot WiFi networks so that with proper device operation, specific attacks on the wireless network will be distinguished, such as deactivation attack, Service Set Identifier (SSID) attack, Address Resolution Protocol (ARP) spoofing, Transmission Control Protocol Synchronize (TCP SYN) authentication and identify the denial. At the same time, collected data is sent to the EIDS system database for processing.

WiFi Honeypot supports a wide range of IEEE 802.11 protocols from IEEE 802.11b to IEEE 802.11ac for data transfer in the available frequency bands of 2.4 GHz and 5 GHz. In this design, the device provides its resources to an attacker at a virtual access point. As shown in Fig. 3, the WiFi Honeypot tool shows the RF environment after setting up and configuring for sniffing and standby mode.

D. Conpot

Honeypot Conpot frameworks are sources of data applications that are used explicitly in industrial control and processing equipment designed to collect anomalous data, detect logs used by unauthorized users, and trap attackers. Honeypot ICS/SCADA Conpot has unique industrial protocols to check the performance of an attacker when logging in. The Conpot is programmed to support all ports used in the industry, such as Modbus, SNMP, HTTP, IP, etc. It can easily encourage attackers to work with it and prevent them from identifying themselves as an industrial fake control process server.

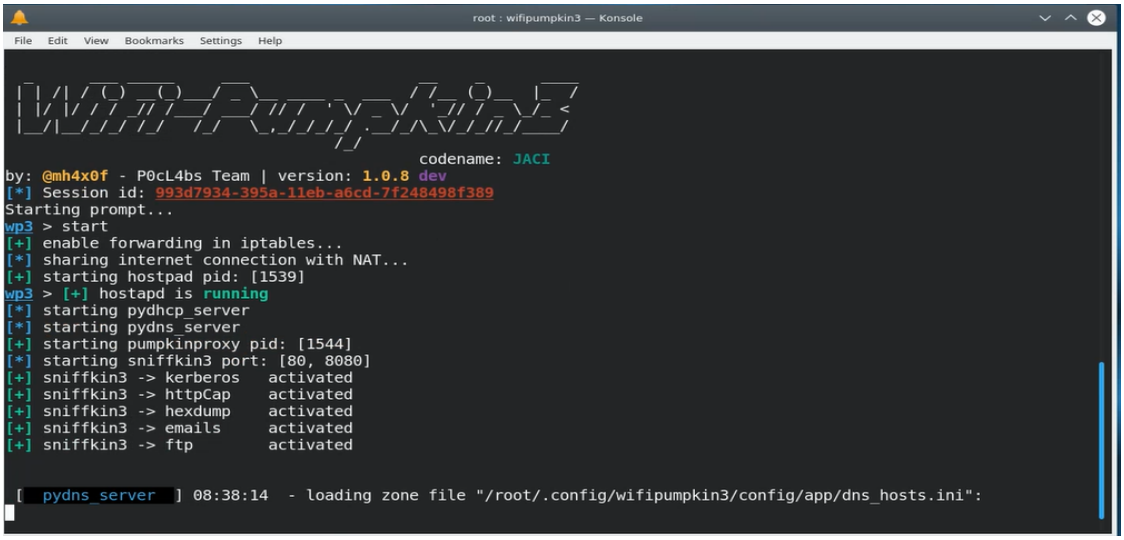


Fig. 3. WiFi Honeypot Sniffing network.

5. Machine Learning

Security systems such as anti-virus, firewall, and IDS are slow in detecting attacks and cause error detection rate, manifesting itself in False Positive (FP) and False Negative (FN), resulting in misdiagnosis, which makes it possible for industrial systems to become vulnerable. Therefore, it is difficult for network administrators to identify new methods of attack. An option for detecting attackers is to use a Honeypot along with ML. Since all activities done by attackers are recorded by Honeypot and will be sent to the EIDS database to store, they are suitable for analysis in ML systems. In ML, classification of data into normal and abnormal classes can improve the diagnosis speed of abnormal situations and the system issues and announces the necessary warning to the system administrator online with a high percentage coefficient and the lowest error detection.

ML gives more choice components in analyzing the behaviors of attackers. It has outstanding performance compared to manual and hardware analysis such as traditional techniques, and it is highly accurate in dealing with Big data. Mean Squared Error (MSE) index makes a distinction between valuation and evaluation values. The error rate decreases as it gets nearer to zero. A function is used in the precision index formula (7).

Formula 10 parameters are as defined:

$$MSE = \frac{1}{n}\sum_{i=1}^{k}\left|x_i - y_i\right|^2 \qquad (7)$$

$$x_i = (x_1, x_2, x_3, ..., x_n), y_i = (y_1, y_2, y_3, ..., y_n)$$

Furthermore, making special adjustments to learning methods makes it possible to automatically display data from raw data and output results to ML algorithms to detect real-time intrusion. A distinctive feature of ML is its DL structure, which, compared to traditional ML, involves several hidden layers. Traditional ML models include one layer. Therefore, its conventional models are called Shallow Learning (SL) models. A simple flowchart about the classification of conventional SL and DL is shown in Fig. 4.

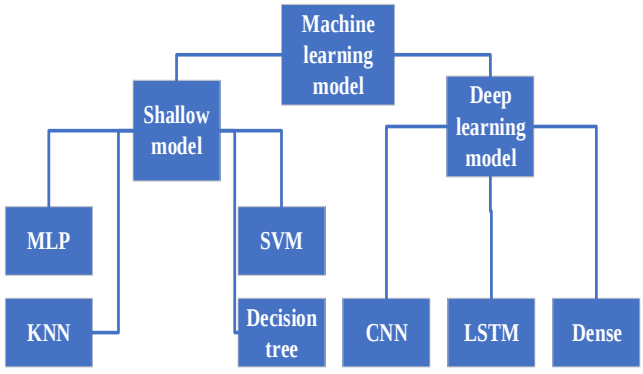


Fig. 4. Used ML algorithms in the proposed scheme.

E. Support Vector Machine (SVM)

In the SVM algorithm method, each data item's classification will be plotted to point in the next n space that n will have multiple EIDS attributes. In this way, the size of each character will have a certain number of coordinates. The properties of variables will be plotted in a two-dimensional space where each point has two

coordinates known as support vectors. By entering this algorithm, the data, given that they will be in their unique coordinates, will form groups that can be classified with a line between these groups. This line will organize two groups. where x Backup vectors, w Multiple-membered vectors, b numerical constant, and $\frac{1}{2}\|w\|^2$ The separation between the superheroes. Modeling of the support vector machine mathematically (8).

$$y = Min\frac{1}{2}\|w\|^2 \qquad (8)$$

$$s.t.$$

$$y_i\left(w.x_i - b\right) \geq 1 \qquad \forall 1 \leq i \leq n$$

F. Multi-Layer Perceptron (MLP)

In MLP, inputs are not directly connected to the output. Instead, there is a middle layer for this. Like the first layer, this layer does the same thing as multiplying the values in the weights and then adding them to the bias value. The last line with a more complex pattern can be discovered using the middle layer, also known as the hidden layer.

G. K-Nearest Neighbor (KNN)

In the KNN algorithm, each new sample's distance must first be compared with the previous samples. To do this, comparing and subtracting each of the unknown sample features from the same features found in an old sample seems to be necessary. Different dimensions are compared on a peer-to-peer basis. Then the results of these reductions are added. K In the k-nearest-neighbor algorithm is the number of nearest neighbors which used to vote on a data sample's status belonging to existing classes. The choice of the parameter k in the k-nearest neighbor algorithm is significant. As the value of k increases, the boundaries of the categories become smoother. The case is then allocated to the nearest neighbor's class [31]. One of the three, four, or five methods are used to calculate the distance function. The parameters for Formulas 9, 10 and 11 are as follows:

Distance function in Euclid (9).

$$y = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \qquad (9)$$

$$x_i = \left(x_1, x_2, x_3, ..., x_n\right), y_i = \left(y_1, y_2, y_3, ..., y_n\right)$$

the Manhattan distance formula (13).

$$y = \sqrt{\sum_{i=1}^{k}\left|x_i - y_i\right|} \qquad (10)$$

Minkowski distance function (14).

$$y = \left(\sum_{i=1}^{k}\left(\left|x_i - y_i\right|\right)^p\right)^{\frac{1}{p}} \qquad p = 1,2,3,... \qquad (11)$$

H. Decision Tree (DT)

The decision tree consists of several nodes and branches that make specimens in such a way that it grows from the root downwards and finally reaches the leaf nodes. In addition, a property identifies each internal or non-leaf node. modeling using mathematics (12).

$$y = \frac{1}{B}\sum_{b=1}^{B} d_b\left(x^{'}\right)$$

$$B = \left(x_1, y_1\right), \left(x_2, y_2\right), ..., \left(x_n, y_n\right) \qquad (12)$$

$$d_b = \{\ \left(x_i, y_i\right)\ \}$$

The parameters for Formulas 1 and 2 are as follows: p_i The likelihood that a sample of data belongs to class I is given by the number of classes in the training set, k. Gini's offensive behavior (13).

$$Gini(Q) = 1 - \sum_{i=1}^{k} p_i^{\,2} \quad p_i = \left(p_1, p_2, p_3, ..., p_n\right) \qquad (13)$$

Gross entropy function (14)

$$Entropy(Q) = -\sum_{i=1}^{k} p_i \log p_i \quad p_i = (p_1, p_2, p_3, ..., p_n) \tag{14}$$

Dense layer

The new generation of ML algorithms, known as DL algorithms, seeks to reduce feature extraction from input data. In these new methods, an attempt is made to extract the feature in the training process and the algorithm itself. This critical point in these networks and their high accuracy in many previously performed tasks by conventional ML algorithms cause this field to grow and develop with incredible intensity.

In the dense layer, fully connected networks are used to classify the traffic of telecommunication and computer networks. These dense networks are made up of layers in which each neuron is connected to all the neurons in its next layer. To build these networks, using a sequential structure to organize the layers seems essential (each layer only relates to the next layer). When using dense layers, the number of neurons in each layer must be first determined. The number of these neurons is a hyperparameter that can help get the right amount by trial and error.

I. One-Dimensional Convolution Neural Network (CNN1D)

The CNN1D with a DL approach is used to develop the IDS function presented in this scheme that gets the data set features properly. The CNN1D model learns a single property in the first layer of convolution and calculates its weight with the defined core size according to the length of the input matrix. Subsequent layers of convolution follow the same logic as the first layer of convolution. Because CNN1D works with filters on input data and can adjust the filters if adequately trained, therefore, it is easier to train CNN1D models with lower initial parameters than other neural networks, and a large number of hidden layers will not be needed because convolutional will be able to control the discovery of hidden layers. The convolutional 1D layer receives the network traffic data packets as an input vector of size where signifies features, and cl denotes the class label for the dataset. as a result of a collection of features f, a new feature map h_i is obtained as a feature map.

$$kl = [kl_1, kl_2, ..., kl_{n-f+1}] \tag{15}$$

And the convolution process.

$$h_i = f(h_{i-1} \otimes w_i = b_i) \tag{16}$$

Extended Short-Term Memory networks (LSTM)

LSTM uses repetition to remember information for long periods by learning long-term dependencies. In a typical neural network, all inputs and outputs are independent of each other, but this idea can be terrible in many cases. For example, suppose a person is trying to predict the next word in a sentence. If the network cannot learn the relationships between the terms, it cannot predict the next word correctly. So, to expect the next time step, it is necessary to update the weights in the network, which requires preserving information of the initial time steps, but LSTMs can learn these long-term dependencies correctly.

Three world-famous datasets are used: NSL-KDD dataset, CIC-IDS-2017 dataset, and Kyoto 2006 dataset explained in the following sections.

J. NSL-KDD

The NSL-KDD data has one test data and one train data, which are network records. The NSL-KDD version has 43 features; 41 are related to incoming traffic, the normal tag or attack, and the other items related to traffic intensity points.

The most important thing in this benchmark database is the attribute tag that specifies the normal or attacks label. The attribute tag tells whether this record is a normal record or an attack record, and all the records in these attributes are data. Also, there are more attacks on test data, and there are cases of unknown attacks but do not deal with this data in this way. Given the number of attacks mentioned and the normal state, five classes for work are considered. Normal, Dos, R2L, U2R, Probe.

Here are five tags for data. The preprocessing work is expanded, and the algorithms used in the DL and SL discussion are chosen to develop a simple process that makes testing and scaling easier.

K. CIC-IDS 2017

CIC-IDS2017 is a dataset with 78 features and respective class labels that include various 14 diversity of attacks, such as brute-force, Denial of Service (DoS), web attacks, etc. By summarizing the above example, it can be concluded that CICIDS2017 counts eight categories: benign, brute-force, DoS, web attacks, infiltration, botnet, port scan.

L. Kyoto 2006

The Kyoto 2006 dataset is the actual network traffic logs extracted from Honeypot sensors. These logs contain data collected from different types of Honeypots that consists of 23 attributes plus a tag attribute. This data log also includes normal traffic and abnormal traffic (types of attacks).In this data set, several columns are defined as primary columns that consist of known attacks, such as the Identification of all attacks in the label column, detected attacks with the exploit code in the Ashula detection column, the detection of Malware attacks in the Malware detection column, and the attack detection with the IDS firewall in the IDS detection column.

M. Metrics

Methods and criteria are needed to measure accuracy, Recall (R), Precision (P), and F1-Score to evaluate the methods used by ML for the proposed design in this paper to achieve the most optimal model for analyzing data properties. Therefore, the following criteria used in this article are briefly explained, along with the relevant formula and equations.

1) Accuracy

The accuracy parameter expresses the number of correct predictions made by the category divided by the number of total predictions made by the same category. That is the ratio of accurate diagnoses True Positive (TP) + True Negative (TN) to the total data that included TP + TN + False Positive (FP) + False Negative (FN). This criterion is very effective for many real-world classification problems because it considers both unintended data (denominator) and identifying data (deduction form) as equation 17. The goal of the proposed method is to reach accuracy=1 or 100%.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{17}$$

$$far = \frac{1}{2}\left(\frac{FP}{FP + TP} + \frac{FN}{FN + TN} \right) \tag{18}$$

2) Recall (R)

Accuracy parameter is not suitable for unbalanced data, data whose number of positive and negative labels in the real world are very different numerically. Many real-world issues fall into this category. This significant difference between different data sets makes the accuracy criterion inefficient. Therefore, a more objective benchmark is needed for measuring the proposed classification algorithms' accuracy and efficiency. In such cases, it is better to focus on the number of TP identified as the total number of positive samples. The R parameter for this purpose is defined as equation 19.

$$R = \frac{TP}{TP + FN} \quad (19)$$

3) Precision (P)

If any positive sample in the R formula cannot be detected, its fraction becomes zero, which indicates that the proposed model is weak and can therefore be quickly rejected. To solve this problem, in addition to the retrieval criterion, another benchmark is defined called Precision, equal to the number of TP diagnostic samples to the total positive samples declared as equation 20 to consider the number of FP.

$$P = \frac{TP}{TP + FP} \quad (20)$$

4) F1-Score

If a combination of these two criteria, R and P, can be obtained for measuring classification algorithms, it would be more appropriate to focus on one criterion instead of examining the two simultaneously. An average of the two as a new criterion can be used in order to raise the arithmetic mean. Therefore, the usual average of the two R and P measures is considered a criterion for high R and low P (or vice versa). In that case, the normal numerical average will be accepted if the proposed algorithm does not get a passing score.

According to equation 21, this average harmony for the two values of R and P is known as F1-Score, which according to the above procedure, is equal to:

$$F1 = 2\frac{P \times R}{P + R} \quad (21)$$

6. Discussion On The Eids

The EIDS presented architecture in Fig. 1 has used several different interconnected industrial networks that form a comprehensive industrial network. Each industrial network has its own identifiers that nodes outside the network must be known in order to be able to send data. Furthermore, there are other networks between the source network and the destination network, and there is more than one route to send data from the source to the destination. Therefore, this extensive network nodes' connection with each other is not as simple as the connections of nodes within a network.

To plan an early intrusion framework, an arrange plan, agreeing to Fig. 1, was utilized. An Industrial switch, used in Fig. 1, sets all the connections in the comprehensive industrial network because the industrial switch has the ability of interfacing users` systems to computers, servers, and IP-based broadcast communications.

The EIDS framework is designed as it is within the instruction discovery mode to be prepared to distinguish assaults on only fake devices. In Fig. 5, a sample of used signatures for anomaly discovery using fake devices created by Honeypot sensors is shown.

```
snort -i eth0 -u snort -g snort -c /etc/snort/snort.conf -l /var/log/snort/


        snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 -l /var/log/snort/
        systemctl status snortd.service
        systemctl status barnyard2.service
=====================================================

#alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001; rev:001;)
#alert icmp any any -> $HOME_NET any (msg:"Possible Nmap ping sweep"; dsize:0; sid:1000005; rev:1;)
alert icmp any any -> 192.168.43.212/24 any (msg: "NMAP ping sweep Scan"; dsize:0;sid:10000004; rev: 1;)
alert tcp any any -> 192.168.43.212/24 6666 (msg: "NMAP TCP Scan";sid:10000005; rev:2; )
alert tcp any any -> 192.168.43.212/24 6666 (msg:"Nmap XMAS Tree Scan"; flags:FPU; sid:1000006; rev:1; )
alert tcp any any -> 192.168.43.212/24 6666 (msg:"Nmap FIN Scan"; flags:F; sid:1000008; rev:1;)
alert tcp any any -> 192.168.43.212/24 6666 (msg:"Nmap NULL Scan"; flags:0; sid:1000009; rev:1; )
alert udp any any -> 192.168.43.212/24 any ( msg:"Nmap UDP Scan"; sid:1000010; rev:1;)


============================================================
```

Fig. 5. Vital Rules in IDS Snort to monitor Honeypot sensors.

Fig. 6 shows the steps for diagnosing anomalies using ML in the logs stored by the databases presented in section VID, which can help system administrators to see the results of detecting anomalies in this designed application in the shortest time and with the least error. The advantage of such an early detection program with ML is to improve system performance in EIDS to prevent intruders from infiltrating key OSs prior to sabotage by such attackers. Therefore, SL and DL in ML are used to assist the EIDS hardware and software detection system to improve performance and reliability to identify and analyze intruders, as shown in Fig. 6. This system offers learning models for processing and extracting implemented results.

```
Using TensorFlow backend.
Reading File: 24 statistical features_test.xls
k-NN Running...
Decision Tree Running...
RandomForest Running...
MLP Running...
KMeans Running...
Dense Running...
CNN1D Running...
Autoencoder LSTM Running...
[28/Oct/2020 12:28:15] "POST /myapp/ HTTP/1.1" 200 17704
Not Found: /myapp/img/spin.svg
[28/Oct/2020 12:28:15] "GET /myapp/img/spin.svg HTTP/1.1" 404 2227
[28/Oct/2020 12:28:15] "GET /static/img/2.png HTTP/1.1" 200 29705
[28/Oct/2020 12:28:15] "GET /static/img/1.png HTTP/1.1" 200 15061
[28/Oct/2020 12:28:15] "GET /static/img/3.png HTTP/1.1" 200 118485
[28/Oct/2020 12:28:15] "GET /static/img/4.png HTTP/1.1" 200 32591
[28/Oct/2020 12:28:15] "GET /static/img/5.png HTTP/1.1" 200 34058
[28/Oct/2020 12:28:15] "GET /static/img/6.png HTTP/1.1" 200 32756
[28/Oct/2020 12:28:15] "GET /static/img/8.png HTTP/1.1" 200 16689
[28/Oct/2020 12:28:15] "GET /static/img/7.png HTTP/1.1" 200 18315
[28/Oct/2020 12:28:16] "GET /static/img/9.png HTTP/1.1" 200 20565
[28/Oct/2020 12:28:16] "GET /static/img/10.png HTTP/1.1" 200 17084
[28/Oct/2020 12:28:16] "GET /static/img/11.png HTTP/1.1" 200 20612
[28/Oct/2020 12:28:16] "GET /static/img/12.png HTTP/1.1" 200 12625
```

Fig. 6. Schematic command program of an ML system designed to process and extract results implemented under the Python Web application.

The rules utilized in designing an EIDS based on ML are used for early attack detection in industrial networks. Several rules have been written to distinguish between a standard package, an unknown package, and an attack package to detect early attacks properly. As the IDS Snort task in the designed EIDS matches the data packets to the router's Iptables pattern and determines whether the containers are operating as instructed, this mode controls the

data packets throughout the incoming network traffic according to the available sensors. Therefore, the EIDS application reports any abnormal traffic instantaneously, as can be seen in the program discussed, which displays abnormal network traffic status according to the code written in those programs. The EIDS administrator takes contingency measures by observing the current status of the reported IPs, and if the administrator wants to know about the intruder's behavior, the mentioned quick process from the attacker's behavior in Fig. 7 can be an excellent solution to decide on the mentioned conditions. One of the advantages of using the proposed network monitoring system is achieving the duration of DDOS attacks with destination IP addresses known as attackers. The design is significant, simple, and user-friendly, and the use of such an advanced system increases the security of industrial control systems in new generation industrial facilities.

```
print("Duration of DDOS Attack by Destination IP Address")
print(tmpDF.groupby("Destination IP Address")["Duration"].sum())
tmpDF.groupby("Destination IP Address")["Duration"].sum().plot(kind='pie', figsize = (12, 12))
```

```
Duration of DDOS Attack by Destination IP Address
Destination IP Address
fd12:36df:4a50:0324:7d0a:2779:6073:4f8f    0.00
fd12:36df:4a50:1218:7ddb:2788:6028:3342    0.00
fd12:36df:4a50:16bb:7d74:2775:6055:037b    2.23
fd12:36df:4a50:2000:7d7e:27a0:606a:13a2    0.00
fd12:36df:4a50:2110:7a2a:00d7:5f46:36b5    0.00
                                          ...
fd12:36df:4a50:ecc2:7db4:27e1:6042:39a8    0.00
fd12:36df:4a50:f0d0:7d35:2739:60f0:0b49    0.00
fd12:36df:4a50:f569:7de5:2781:60d3:1e7e    0.00
fd12:36df:4a50:f7bb:7d17:279a:6083:1b71    2.07
fd12:36df:4a50:f890:7d5e:27de:60d2:0fd9    0.00
Name: Duration, Length: 78, dtype: float64
```

Fig. 7. Duration of DDOS attack by destination IP Address.

As shown in Fig. 8, Conpot Honeypot sensors are utilized as deceiver Industrial PLCs and equipment sensors. The assignment of these imitator frameworks in computer, media transmission, and processing is to play down network attacker's detection time. That's why accessing these frameworks has become facilitated to create them appealing to assailants and information. For example, Fig. 8 illustrates a deceiver Conpot recorder framework on the Ubuntu OS framework called fake Conpot-1.

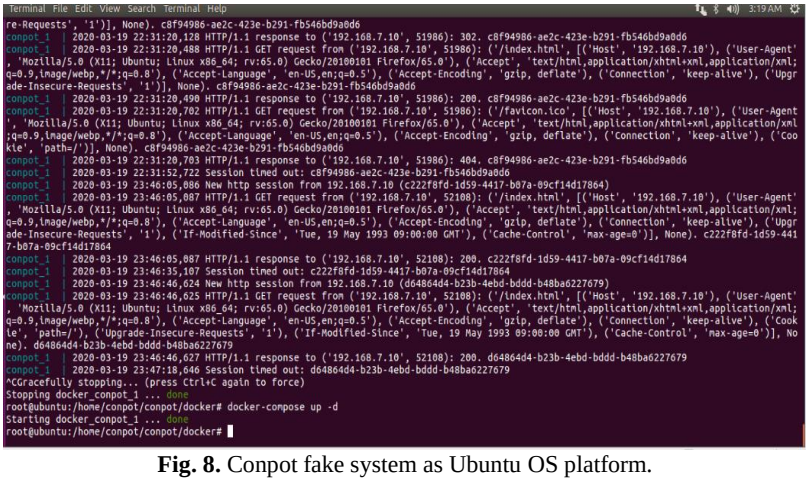


Fig. 8. Conpot fake system as Ubuntu OS platform.

In Fig. 9, as shown, immediately after detecting abnormal traffic, attacker information is sent by EIDS to the system administrator's monitor screen for display. The display of attacking information depends on the definition of our rules, and this is a significant design advantage. Attack type, IP of origin, and destination of the attack are also displayed.

```
[root@localhost rules]# snort -A console -q -u snort -g snort -c /etc/snort/snort.conf
-i eth0 -l /var/log/snort/
01/09-18:10:31.961001  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.1
68.50.26:50768 -> 192.168.50.68:6666
01/09-18:10:31.961604  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.1
68.50.26:50768 -> 192.168.50.68:6666
```

Fig. 9. An alert in the EIDS monitoring screen.

EIDS has several advantages, such as showing cautions as logs in its program. To group these cautions, comprehensive monitoring needs to be used. In addition to the EIDS project program based on ML, another comprehensive monitoring system compatible with the EIDS is the Basic Analysis and Security Engine (BASE). All the recall work is recorded by the assailants on the trap Conpot frameworks. At this time, early interruption location cautions are sent by the EIDS application at the side of all Information determinations to BASE. Fig. 10 shows the appearance of the framework along with the taken information attack from Honeypot sensors in the BASE screen.

Sensors/Total: 1 / 1
Unique Alerts: 7
Categories: 1
Total Number of Alerts: 25855

- Src IP addrs: 14
- Dest. IP addrs: 252
- Unique IP links 282
- Source Ports: 50
  - TCP (35) UDP (15)
- Dest Ports: 403
  - TCP (1) UDP (402)

Fig. 10. BASE Screen.

7. Results

Modeling the proposed EIDS is tried to comprehensively cover all available ICSs such as RF systems, PLC, temperature sensors, pressure sensors, flow sensors, position valves, actual valves, and types of actuators such as control valves, actuators with electric motors, etc. Accordingly, in this study, for all the mentioned industrial control facilities, simulation was done. Due to the comprehensiveness of the proposed work, modeling and application programing were studied and performed in several different telecommunication areas, industrial computer, control processing, and instrumentation. Simulations were performed in several other regions due to the work scope of the simulation; for example, on industrial wireless networks, it was utterly different from simulations in the field of industrial PLCs.

In the real industrial environment, the proposed system must detect the attackers' attacks early. So, simulation and design work had to be done for several different areas mentioned simultaneously. Therefore, these systems' incoming and outgoing traffic logs were collected in existing facilities, which are practically in several various networks with other protocols, in a comprehensive database as the EIDS system database.

The system shows suspicious attack traffic on the BASE screen and the screen designed for this research, which is mentioned in section VIF at the same time as detection. If necessary, the industrial facility security system manager, by analyzing the data received from the attacker by ML, examines all the attacker's behavior and makes a decisive decision for the attacker before the attacker has enough time and opportunity to sabotage the control systems of industrial facilities.

So, the incoming and outgoing traffic logs collected from Honeypot's extensive network of sensors are real in this simulation. The labeling of this data is also privatized and applied using a creative idea. Also, labeling network logs is scalable because with operational experience in the laboratory simulation environment, as Fig. 11, the strengths of the data infrastructure obtained from the sensors were created and established according to the dataset's needs. In Fig. 11, detected normal (valid source & destination) and anomalies (invalid source & destination) traffic with Python simulation application EIDS project from recorded real EIDS database are given as well as measured distinguished amount of normal and attack logs shown as a percentage in the pie chart in the real-time.

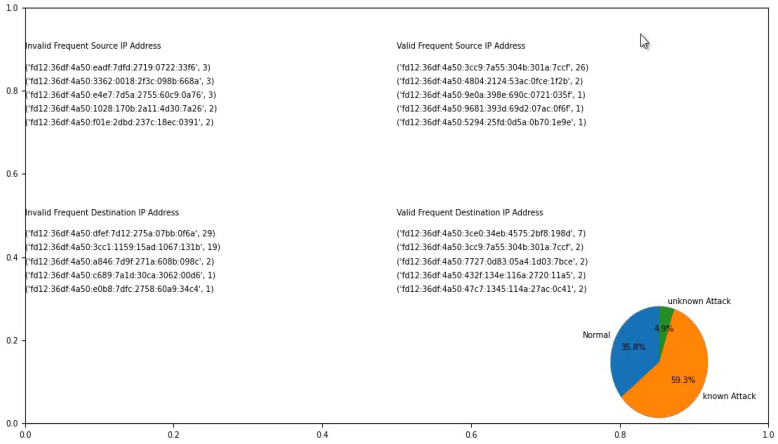


Fig. 11. Detected normal (valid source & destination) and anomalies (invalid source & destination) traffic with Python simulation application EIDS project from recorded real EIDS database and measured amount of normal and abnormal attacks are given in percentage in the pie chart.

The EIDS database is needed to be correctly labeled, as far as possible, by comparing the simulation results obtained from other datasets mentioned in this field with the results obtained from this study. Complete data labeling, accuracy, and scientific work methods are accessible in the EIDS study. In this research, various approaches and strategies for collecting correct data have been studied. Besides, repetitive records in the test cause confusion in the detection rate of the evaluation result in duplicative documents. All duplicate records of

the entire train and test data were deleted to solve this problem, and only one copy of each document has remained. The preprocessing is adjusted according to the simulation needs in the laboratory relative to the obtained results. This section performs a series of experiments on the datasets to demonstrate the correctness of the EIDS research project dataset in this paper.

The performance of early detection of the proposed EIDS system using stored logs in the database using ML compared with other significant datasets in the world, which were mentioned in sections Ⅵ, and their results were obtained in section Ⅷ were analyzed, processed, and reviewed, and the results obtained from SL and DL algorithms for EIDS indicate the success of the IDS in its early detection. Besides, EIDS performed in all the algorithms under the same conditions as with the other datasets.

In Fig. 12 (a), and 12 (b), respectively, the results obtained for detecting anomalies traffic in the EIDS database using different ML types have been performed on the mentioned algorithms. The two crucial criteria, accuracy and F1-Score for EIDS, show that they have more satisfying detection results from anomalies with high accuracy compared to other datasets. Therefore, accuracy and F1-Score have an impressive and acceptable status and show the efficiency and improvement of the proposed method compared to other datasets.

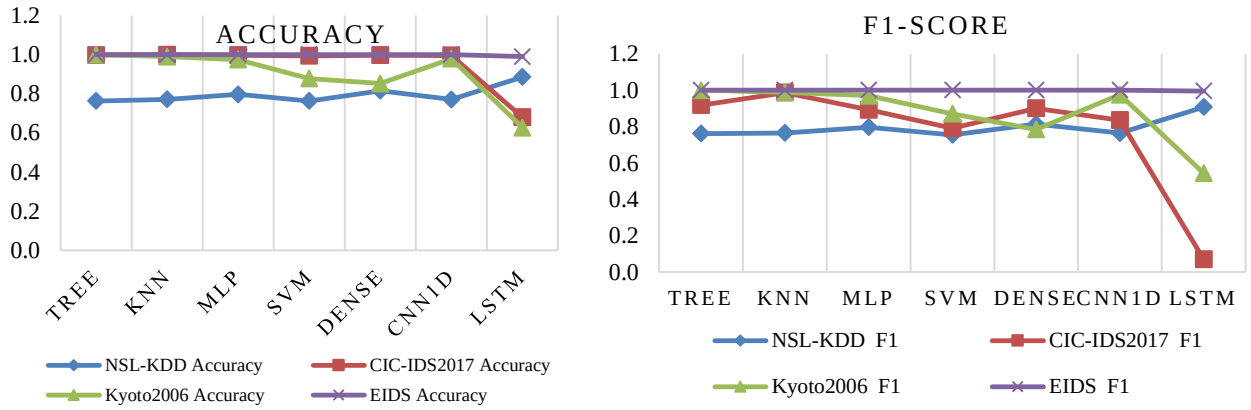
Also, in Tab. 1 and 2, respectively, for the two essential criteria accuracy and F1-Score, the improvement rate of the obtained results is given in percentage for detecting traffic anomalies of the proposed EIDS database compared to the three datasets mentioned in this research. This improvement is so significant for EIDS that the designed method for this study can detect the intrusion of abnormal traffic and shows high accurate performance. Therefore, this design can be used in industrial facilities with high reliability.

Tab. 1. The improvement rate of the obtained results for accuracy is written in percentage for detecting traffic anomalies of the proposed EIDS database compared to the three other datasets mentioned in this research.

Dataset Method	NSL-KDD Accuracy	CIC-IDS2017 Accuracy	Kyoto2006 Accuracy
Tree	31.00%	0.20%	0.05%
KNN	29.59%	0.02%	0.99%
MLP	25.50%	0.31%	2.64%
SVM	31.06%	0.60%	13.97%
Dense	22.66%	0.29%	17.35%
CNN1D	29.80%	0.46%	2.10%
LSTM	11.67%	45.61%	58.03%

Tab. 2. The improvement rate of the obtained results for F1-Score is written in percentage for detecting traffic anomalies of the proposed EIDS database compared to the three other datasets mentioned in this research.

Dataset Method	NSL-KDD F1-Score	CIC-IDS2017 F1-Score	Kyoto2006 F1-Score
Tree	31.39%	8.94%	0.06%
KNN	30.75%	1.45%	1.18%
MLP	25.62%	12.10%	3.15%
SVM	32.58%	26.49%	14.89%
Dense	23.00%	11.09%	27.49%
CNN1D	30.92%	19.82%	2.59%
LSTM	9.67%	1316.56%	82.97%



(a) Accuarey (b) F1-Score

Fig. 12. The extent to which anomaly traffic detection is detected by the accuracy (a) and the F1-Score (b) index in the EIDS database compared to other datasets using ML algorithms in the form of line charts.

Conclusion

This paper first introduces an architecture called EIDS that uses the Honeypot method, which utilizes data sources for early detection using ML focusing on shallow and deep learning techniques. According to the considered classification, different data sources such as logs, packets, flows, and different sessions are being analyzed for learning algorithms. To produce a suitable database, simulation of DDoS attacks, sniffing attacks, Port Scanner attacks, etc. on sensors with Ubuntu OS, Centos OS, Raspberry OS, Windows OS, PLC frameworks, WiFi, Router OS, MikroTik OS, MikroTik switches, and other required tools and equipment were performed. The EIDS system, using expanded sensors such as Conpot ICS/SCADA Honeypot, WiFi Honeypot, Cowrie Honeypot, and Kippo Honeypot, successfully stored the attack logs in a laboratory environment for industrial installations and sent a copy of the attack data and logs directly to EIDS database. In fact, it offers a special solution based on Honeypot and a combination of ML algorithms for modeling and forecasting to identify and classify the characteristics such as normal and abnormal (suspicious) data. Simultaneously with these steps, the effects of attacks on all utilized sensors in this research were also displayed in the web of EIDS project and BASE monitoring to the system administrator. By analyzing the received data from the sensor traffic, the system administrator could easily analyze the attacker's behavior in real-time in the EIDS based on the industrial Honeypot designed for this research.

References

[1] Khoda, M. E., Imam, T., Kamruzzaman, J., Gondal, I., & Rahman, A. (2019). Robust malware defense in industrial IoT applications using machine learning with selective adversarial samples. IEEE Transactions on Industry Applications, 56(4), 4415-4424.

[2] Al Hasnain, F., Sahami, A., & Kamalasadan, S. (2021). An Online Wide-Area Direct Coordinated Control Architecture for Power Grid Transient Stability Enhancement Based on Subspace Identification. IEEE Transactions on Industry Applications, 57(3), 2896-2907.

[3] Pei, C., Xiao, Y., Liang, W., & Han, X. (2020). Pmu placement protection against coordinated false data

injection attacks in smart grid. IEEE transactions on industry applications, 56(4), 4381-4393.

[4] Pashaei, A., Akbari, M. E., Lighvan, M. Z., & Teymorzade, H. A. (2020, June). Improving the IDS performance through early detection approach in local area networks using industrial control systems of honeypot. In 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe) (pp. 1-5). IEEE.

[5] Pliatsios, D., Sarigiannidis, P., Liatifis, T., Rompolos, K., & Siniosoglou, I. (2019, September). A novel and interactive industrial control system honeypot for critical smart grid infrastructure. In 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (pp. 1-6). IEEE.

[6] Yamin, M. M., Katt, B., Sattar, K., & Ahmad, M. B. (2019, March). Implementation of insider threat detection system using honeypot based sensors and threat analytics. In Future of Information and Communication Conference (pp. 801-829). Springer, Cham.

[7] Zhao, C., & Qin, S. (2017, December). A research for high interactive honepot based on industrial service. In 2017 3rd IEEE International Conference on Computer and Communications (ICCC) (pp. 2935-2939). IEEE.

[8] Pashaei, A., Akbari, M. E., Lighvan, M. Z., & Charmin, A. "Detection Anomaly of Network Datasets With Honeypots at Industrial Control System". Journal of Artificial Intelligence in Electrical Engineering, 2023.

[9] Matin, I. M. M., & Rahardjo, B. (2019, November). Malware detection using honeypot and machine learning. In 2019 7th International Conference on Cyber and IT Service Management (CITSM) (Vol. 7, pp. 1-4). IEEE.

[10] Vanhoenshoven, F., Nápoles, G., Falcon, R., Vanhoof, K., & Köppen, M. (2016, December). Detecting malicious URLs using machine learning techniques. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-8). IEEE.

[11] Siniosoglou, I., Efstathopoulos, G., Pliatsios, D., Moscholios, I. D., Sarigiannidis, A., Sakellari, G., ... & Sarigiannidis, P. (2020, July). NeuralPot: An Industrial Honeypot Implementation Based On Deep Neural Networks. In 2020 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-7). IEEE.

[12] Vishwakarma, R., & Jain, A. K. (2019, April). A honeypot with machine learning based detection framework for defending IoT based Botnet DDoS attacks. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1019-1024). IEEE.

[13] Kwon, D., Natarajan, K., Suh, S. C., Kim, H., & Kim, J. (2018, July). An empirical study on network anomaly detection using convolutional neural networks. In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) (pp. 1595-1598). IEEE.

[14] Nayyar, S., Arora, S., & Singh, M. (2020, July). Recurrent Neural Network Based Intrusion Detection System. In 2020 International Conference on Communication and Signal Processing (ICCSP) (pp. 0136-0140). IEEE.

[15] A. Pashaei, M. E. Akbari, M. Zolfy Lighvan, and A. Charmin, "A Honeypot-assisted Industrial Control System to Detect Replication Attacks on Wireless Sensor Networks", Majlesi Journal of Telecommunication Devices, Vol. 11, No. 3, pp. 155-160, 2022.

[16] Mashima, D., Chen, B., Gunathilaka, P., & Tjiong, E. L. (2017, October). Towards a grid-wide, high-fidelity electrical substation honeynet. In 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm) (pp. 89-95). IEEE.

[17] Dalamagkas, C., Sarigiannidis, P., Ioannidis, D., Iturbe, E., Nikolis, O., Ramos, F., ... & Tzovaras, D. (2019, June). A survey on honeypots, honeynets and their applications on smart grid. In 2019 IEEE Conference on Network Softwarization (NetSoft) (pp. 93-100). IEEE.

[18] Shi, L., Li, Y., Liu, T., Liu, J., Shan, B., & Chen, H. (2019). Dynamic distributed honeypot based on blockchain. IEEE Access, 7, 72234-72246.

[19] Pashaei, A., Akbari, M. E., Lighvan, M. Z., & Charmin, A. (2022). Early Intrusion Detection

System using honeypot for industrial control networks. Results in Engineering, 16, 100576.

[20] Nursetyo, A., Rachmawanto, E. H., & Sari, C. A. (2019, October). Website and network security techniques against brute force attacks using honeypot. In 2019 Fourth International Conference on Informatics and Computing (ICIC) (pp. 1-6). IEEE.

[21] Baykara, M., & Das, R. (2018). A novel honeypot based security approach for real-time intrusion detection and prevention systems. Journal of Information Security and Applications, 41, 103-116.

[22] Ziaie Tabari, A., & Ou, X. (2020, October). A Multi-phased Multi-faceted IoT Honeypot Ecosystem. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (pp. 2121-2123).

[23] Yang, H., Cheng, L., & Chuah, M. C. (2019, June). Deep-learning-based network intrusion detection for SCADA systems. In 2019 IEEE Conference on Communications and Network Security (CNS) (pp. 1-7). IEEE.

[24] Gao, J., Gan, L., Buschendorf, F., Zhang, L., Liu, H., Li, P., ... & Lu, T. (2020). Omni SCADA intrusion detection using deep learning algorithms. IEEE Internet of Things Journal, 8(2), 951-961.

[25] Perez, R. L., Adamsky, F., Soua, R., & Engel, T. (2018, August). Machine learning for reliable network attack detection in SCADA systems. In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 633-638). IEEE.

[26] Sheng, C., Yao, Y., Fu, Q., & Yang, W. (2021). A cyber-physical model for SCADA system and its intrusion detection. Computer Networks, 185, 107677.

[27] Khan, I. A., Pi, D., Khan, Z. U., Hussain, Y., & Nawaz, A. (2019). HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. IEEE Access, 7, 89507-89521.

[28] Qian, J., Du, X., Chen, B., Qu, B., Zeng, K., & Liu, J. (2020). Cyber-physical integrated intrusion detection scheme in SCADA system of process manufacturing industry. IEEE Access, 8, 147471-147481.

[29] Bulle, B. B., Santin, A. O., Viegas, E. K., & dos Santos, R. R. (2020, October). A Host-based Intrusion Detection Model Based on OS Diversity for SCADA. In IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society (pp. 691-696). IEEE.

[30] Al-Asiri, M., & El-Alfy, E. S. M. (2020). On Using Physical Based Intrusion Detection in SCADA Systems. Procedia Computer Science, 170, 34-42.

[31] A. Pashaei, M. E. Akbari, M. Zolfy Lighvan, and A. Charmin, " Honeypot Intrusion Detection System using an Adversarial Reinforcement Learning for Industrial Control Networks", Majlesi Journal of Telecommunication Devices, Vol. 12, No. 1, 2023.