An Ensemble-Based Machine Learning-Envisioned Intrusion Detection in Industry 5.0-Driven Healthcare Applications

Mohammad Wazid, *Senior Member, IEEE*, Jaskaran Singh, *Student Member, IEEE*,
Ashok Kumar Das, *Senior Member, IEEE*, and Joel J. P. C. Rodrigues, *Fellow, IEEE*

***Abstract*—With the emergence of advanced technology and biotechnology, medical institutions like hospitals are increasingly relying on smart devices to create an efficient ecosystem. The Industry 5.0-driven healthcare system has started its focus on personalizing products/services having unique and special needs for patients with various diseases. In the current scenario of smart healthcare, we need to consider a human-centric solution that induces the Internet of Things (IoT), Internet of Medical Things (IoMT) and Artificial Intelligence (AI). The fifth industrial revolution, which is being pushed by Industry 5.0, is noted for its ability to meet the customized needs of both patients and healthcare providers. It offers a product to patients and medical professionals in accordance with their unique needs. The Industry 5.0-driven healthcare system has a vast variety of applications, such as remote consultation, routine health monitoring and support, critical care and alert, etc. However, it also suffers from various security and privacy-related issues as various cyber attacks can be launched on the Industry 5.0-driven healthcare system. Therefore, we need a robust security mechanism to protect sensitive healthcare data and other resources. The machine learning (ML)/deep learning (DL) model can be effective to some extent, ensemble-based models have emerged as a promising approach for addressing the potential security threats. In this article, we propose a novel ensemble-based ML-envisioned scheme to detect different types of intrusions for the Industry 5.0-driven healthcare system (in short EIDS-HS). We validate the proposed EIDS-HS on a standard data set and evaluate its performance using key performance parameters, such as accuracy, precision, recall, F1-score, and computational complexity. The security analysis of the proposed EIDS-HS proves its security against various possible attacks. Furthermore, EIDS-HS performs better than existing intrusion detection schemes in terms of important performance parameters.**

***Index Terms*—Industry 5.0, smart healthcare, cyber attacks, malware detection, machine learning, security.**

I. Introduction

THE INDUSTRY 5.0-driven healthcare system focuses on the personalizing of products having unique and special needs for patients according to their diseases [1], [2]. Industry 5.0-driven healthcare system has a great impact on the consumer digital ecosystems (especially for patients). In the current scenario of smart healthcare, we need to consider a human-centric solution that induces (the Internet of Medical Things (IoMT)) and Artificial intelligence (AI). The fifth industrial revolution, which is being pushed by Industry 5.0, is noted for its ability to meet the customized needs of both patients and healthcare providers. However, its previous versions were not as human-centric and were insufficient. Industry 5.0 is enabling the health sector to transition from mass customization to mass personalization as they currently pursue mass personalization with a human touch [3]. It offers products and services to patients and medical professionals as per their needs [4], [5]. The industrial revolution refers to the interaction between humans and machines (through AI, IoT) to enhance and speed up operations. The Industry 5.0-driven healthcare system has a vast variety of applications, i.e., remote consultation, routine health monitoring and support, critical care and alert, disease prediction, laboratory test and analysis, etc. [2], [6].

The risk of medical data stealing through intrusions is a significant concern in the healthcare industry. Intrusions can occur through various means, including hacking, phishing, and malware attacks, among others. Once the intruders gain access to medical data, they can use it for various malicious activities, including identity theft, insurance fraud, and financial gain [7]. Medical data theft can also lead to reputational damage for healthcare providers, causing patients to lose trust in their services. To mitigate this risk, healthcare providers need to invest in robust cybersecurity measures, including firewalls, encryption, and access controls. Staff members should also be trained on how to identify and respond to cybersecurity threats [8].

Machine learning can be highly effective in intrusion detection by analyzing large amounts of data in real-time and detecting patterns of malicious activity [9]. Machine learning algorithms can be trained on labelled data sets to identify known malicious activities, such as known types of cyber attacks. However, machine learning can also be effective in

Manuscript received 5 April 2023; revised 24 May 2023; accepted 21 September 2023. Date of publication 25 September 2023; date of current version 26 April 2024. This work was supported in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 313036/2020-9. *(Corresponding author: Ashok Kumar Das.)*

Mohammad Wazid and Jaskaran Singh are with the Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun 248002, India (e-mail: wazidkec2005@gmail.com; jaskaran.jsk2001@gmail.com).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad 500032, India (e-mail: ashok.das@iiit.ac.in).

Joel J. P. C. Rodrigues is with the COPELABS, Lusófona University, 1749-024 Lisbon, Portugal (e-mail: joeljr@ieee.org).

Digital Object Identifier 10.1109/TCE.2023.3318850

1558-4127 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

identifying previously unknown or novel types of intrusions by identifying anomalous behaviour patterns. Machine learning can also help improve the accuracy and efficiency of intrusion detection systems by reducing false positives and false negatives. By continually learning from new data and updating its models, machine learning can adapt to changing attack patterns and reduce the number of false alarms.

The Industry 5.0-driven healthcare system has a vast variety of applications as discussed earlier. At the same time, it also suffers from various security and privacy-related issues as various cyber attacks, i.e., "malware injection, replaying of information, unauthorised healthcare data disclosure and updates, impersonations, credential guessing, denial-of-service (DoS) etc.," are possible [2], [6]. Therefore, we need some security mechanisms to protect sensitive healthcare data and other resources. In machine learning, ensemble models aggregate several different individual models to enhance the system's overall performance [10], [11]. Ensemble models are less susceptible to over-fitting as each individual model in the ensemble has been trained on a different subset of the data. Also, ensemble models can reduce bias in the final predictions by combining the outputs of multiple models, thereby reducing the impact of any single model's limitations. Finally, ensemble models can provide more robust predictions by leveraging the strengths of different individual models to overcome any weaknesses or limitations of a single model. Overall, ensemble models have been shown to be effective in improving the performance of machine learning systems and are widely used in various applications, such as ailment prediction, speech recognition and intrusion detection [12]. Therefore, the ensemble models-based approach has been utilized in the design of the proposed EIDS-HS.

A. Research Motivation

This work's motivation can be summarized as follows. Intrusion attacks are a constant threat to computer networks and systems, and the need for effective detection and prevention measures is crucial. Intrusion attacks can take many forms, including malware infections, denial of service attacks, and phishing attempts, among others. The nature of these attacks is continually evolving, with hackers developing new and sophisticated techniques to bypass security measures and gain access to sensitive data [13], [14]. As a result, ongoing research into intrusion detection and prevention is essential to stay ahead of these threats and develop effective countermeasures. The cyber attacks on the Industry 5.0-driven healthcare system pose a significant threat to patients' privacy and data security. Cybercriminals often target healthcare organizations because of the sensitivity of the data they possess, which includes confidential patient medical records and personal information. Almost 200,000 systems in 150 countries were impacted by the WannaCry ransomware outbreak in 2017, including healthcare organizations in the United Kingdom. It exploited a vulnerability in Microsoft Windows systems, causing widespread disruption and highlighting the vulnerability of healthcare organizations to cyber threats [13]. The consequences of a cyber attack on the Industry 5.0-driven healthcare system can be devastating, with long-term effects on both the organization and patients. Therefore, effective intrusion detection and prevention measures are critical to protecting healthcare data and maintaining patient privacy [13], [15].

The role of intrusion detection systems (IDS) is crucial in safeguarding the sensitive healthcare data and resources of the Industry 5.0-driven healthcare system from various cyber-attacks [14], [15]. IDS use a range of techniques, including signature-based detection, anomaly detection, and behavior-based detection, to identify and prevent unauthorized access. However, IDS are not foolproof, and hackers are continually developing new methods to evade detection [15]. Therefore, ongoing research and development are necessary to improve the accuracy and effectiveness of IDS and develop new techniques to counter emerging threats [16], [17]. Hence, in this paper, we focus on the designing of an ensemble-based machine learning-envisioned intrusion detection technique.

B. Research Contributions

The research contributions of this paper are given below:

- This paper proposes an intrusion detection scheme that uses an ensemble-based machine learning approach.
- A security analysis of the EIDS-HS is presented to demonstrate its ability to withstand various types of attacks.
- To further verify the security of the EIDS-HS, a formal verification using the Scyther tool is also conducted, which confirms its resilience against different cyber attacks.
- To assess its impact on various performance parameters, a practical demonstration of the EIDS-HS is performed.
- In the comparative study, it is observed that the EIDS-HS performs better than existing intrusion detection schemes.

C. Organization of Paper

The remaining parts of the paper are organized as follows. Section II contains the details of the other similar existing schemes of intrusion detection, which are applicable to smart healthcare. The various system models, i.e., network model and threat model of the proposed EIDS-HS are given in Section III. The details of the proposed EIDS-HS are given in Section IV. Further, the practical implementation of EIDS-HS is provided in Section V. Section VI contains the details of various conducted comparisons of the proposed EIDS-HS and other existing schemes. Then the important security analysis of EIDS-HS is provided in Section VII. Furthermore, the formal security verification using the Scyther tool of EIDS-HS is conducted in Section VIII. Finally, the paper is concluded in Section IX with some concluding remarks and future works.

II. Related Work

In this section, we provide the details of some of the existing schemes of intrusion detection.

The use of machine learning in developing intrusion detection systems has been highly beneficial. Machine learning algorithms have demonstrated the ability to detect and mitigate complex attacks that traditional systems cannot, by

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

analyzing large amounts of data and identifying patterns in real-time [18].

Liu et al. [19] proposes a set sampling technique to address a class imbalance in the intrusion detection paradigm. The technique involves using the KMeans algorithm and continuous attributes of minority samples to generate new instances for the minority class. The approach was evaluated on benchmark datasets such as NSL-KDD and CSE-CIC-IDS2018. Liu et al. [20] implemented a solution that includes a learning algorithm that utilizes a Gaussian mixture model with a distance-based method to maintain clusters of malware classes. In addition, XGBOOST algorithm is used to build an intrusion detection system that produces optimal results. Wan et al. [21] utilized feature selection algorithms and a big data framework for packet preprocessing, merging, and labelling malicious data to reduce data size. A decision tree model was then implemented to induce the introduced detection system.

Li et al. [22] implements federated deep learning mechanisms to detect cyber threats in an industrial setting. The mechanism includes a convolutional neural network (CNN) and gated recurrent unit (GRU) module, which are coupled with the federated learning framework. The combination of these components effectively creates a comprehensive wide inclusion reduction model. Prasse et al. [23] presents an implementation of deep learning for analyzing network traffic, utilizing long short-term memory-based models for HTTP traffic analysis. The proposed protocol is scalable and able to detect new malware based on various features such as domain name, client classifier, and host domain classification. The study conducted an experimental evaluation to demonstrate the mechanism's effectiveness in detecting new malware.

Piskozub et al. [24] targeted a large-scale network comprising 65000 samples and 23 billion network flows for malware detection. The study implemented traffic analysis to detect malware based on statistical features of malware behavior and network flow level data. Abedin and Waheed [25] proposes an intrusion detection system that employs a weighted random forest features selection technique. The proposed system is evaluated on two datasets, namely UNSW-NB15 and Network TON_IoT, using various models. The feature selection technique is utilized to select the most relevant features for the binary classification task. The selected features are given adjusted weights for the two different classes. The proposed system outperforms existing methods in terms of F1 score by electing 20 features with this feature selection algorithm.

Chiramdasu et al. [26] proposes the use of logistic regression-based models for malicious URL reduction, which is one of the most common methods used for phishing and inclusion attacks. The study utilizes a dataset from sources like PhishTank and Kaggle. The proposed method outperforms traditional malicious URL methods. Swarna Priya et al. [27] highlights the effectiveness of deep learning networks when used in conjunction with feature engineering for intrusion detection. This study presents a hybrid approach that integrates feature engineering with deep learning techniques for intrusion detection in the Internet of Medical Things (IoMT) environment. The proposed approach employs a parameter selection tuning algorithm to optimize its performance. The study finds that the proposed approach outperforms existing methods in terms of accuracy and significantly reduces time complexity. The results provide compelling evidence for the effectiveness of the proposed approach in detecting intrusions in IoMT environments.

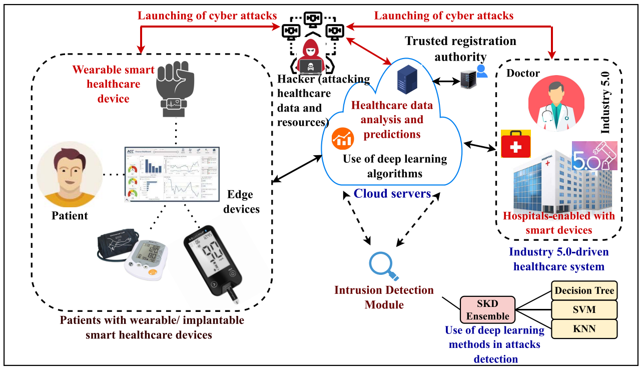

Launching of cyber attacks
Launching of cyber attacks
Trusted registration authority
Wearable smart healthcare device
Hacker (attacking healthcare data and resources)
Doctor
Industry 5.0
Healthcare data analysis and predictions
Use of deep learning algorithms
Patient
Edge devices
Cloud servers
Hospitals-enabled with smart devices
Industry 5.0-driven healthcare system
Intrusion Detection Module
Decision Tree
SKD Ensemble
SVM
KNN
Use of deep learning methods in attacks detection
Patients with wearable/ implantable smart healthcare devices

Fig. 1. Network model of the proposed EIDS-HS.

Immanuel Jeo Sherin and Radhika [28] proposed a stacking-based ensemble that, in addition, to feature extraction, employed stacking to develop a classifier for intrusion detection. Their proposed methodology was tested on the NSL-KDD dataset, where it successfully differentiated between different types of attacks, including DOS, Probe, U2R, and R2L. Esmaeili et al. [29] utilized in the field of intrusion detection, the authors of the study employed Bidirectional Long Short-Term Memory (BiLSTM) networks, in combination with Multi-Layer Perceptron (MLP) models, to detect intrusion attempts on the NSL-KDD dataset. Zou et al. [30] proposed an intrusion detection system based on a decision tree, twin support vector machine, and hierarchical clustering. Hierarchical clustering was used to create the base decision tree classifier, while twin support vector machines were integrated with the base classifier. Ketepalli and Bulla [31] implemented a hybrid intrusion detection scheme by combining a Long Short-Term Memory (LSTM) autoencoder and a random forest feature extraction phase. The LSTM autoencoder was used to learn the underlying patterns in the data, while the random forest was used to select the most relevant features for classification on the NSL KDD dataset.

Most of the schemes, which are discussed here either lack the important performance parameters or insure against various attacks. Therefore, there is great scope for the invention of a new scheme for the detection of intrusions.

III. System Model

The system models, i.e., the network model (overall architecture) of the proposed EIDS-HS and its associated threat model are discussed below.

A. Network Model

The visual representation of the proposed EIDS-HS is given in Figure 1, where we have a scenario of an Industry 5.0-driven healthcare system. In the given scenario, there are patients having wearable/implantable smart healthcare devices,

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

which sense and send the healthcare data to the cloud servers. Over the cloud servers, the healthcare data is analysed and predictions about various illnesses have been made. For this task, various machine learning/ deep learning algorithms are used. The Industry 5.0-driven healthcare system has started its focus on the personalizing of products/ services having unique and special needs for patients with various diseases/ disorders [1]. In the current scenario of smart healthcare, we need to consider a human-centric solution that induces IoT (IoMT in the case of healthcare) and Artificial intelligence (AI). Industry 5.0, which is driving the fifth industrial revolution, is recognized for its capacity to satisfy the specialised requirements of both patients and healthcare professionals. Even though earlier versions weren't as human-centric, they weren't good enough. Industry 5.0 is pressuring the health industry to transition from mass customization to mass personalization since it is currently seeking mass personalization with a human touch. According to their specific requirements, it provides patients and medical professionals with a product. The industrial revolution refers to the interaction between humans and machines (through AI, IoT) to enhance and speed up operations. The Industry 5.0-driven healthcare system has a vast variety of applications, i.e., remote consultation, routine health monitoring and support, critical care and alert, disease prediction, laboratory test and analysis, etc. Though it is applicable in various applications. At the same time, it also suffers from various security and privacy-related issues as various cyber attacks, i.e., "malware injection, replaying of information, unauthorised healthcare data disclosure and updates, impersonations, credential guessing, and denial-of-service (DoS)," attacks are possible. Therefore, we need some security mechanisms to protect sensitive healthcare data and other resources. In the proposed EIDS-HS, the task of intrusion is done by the cloud servers in a joint operation with the proxy servers, hospital servers and deployed capture engines. We have used an ensemble-based ML-envisioned scheme for the detection of different types of intrusions. The cloud servers are the trusted entity of the network as they do the task of intrusion detection. The proxy servers and hospital servers can be treated like the semi-trusted entities of the network. There is a trusted registration authority (a trusted system), which does the task of registration of various entities, i.e., proxy servers, hospital servers and cloud servers.

B. Threat Model

The Dolev-Yao (DY) threat model, which is currently considered the de facto standard, was utilized in the development of EIDS-HS [32]. Two distinct entities may establish communication across an insecure network, according to the DY model (for example, over the Internet). The endpoint entities, such as smart healthcare devices and servers are frequently unreliable. The communications sent via an unprotected network can be intercepted, altered, or deleted by an adversary $\mathcal{A}$, whether passive or active. Also, we have also employed another important adversary model developed by Canetti and Krawczyk (CK) [33]. Here $\mathcal{A}$ can utilise all of the DY model's capabilities. Moreover, $\mathcal{A}$ can also obtain the session states, which are a specific session's credentials and session keys. $\mathcal{A}$ can use a complex power analysis technique to manually seize some smart healthcare devices and gain access to the information stored in their memory [34]. The information gathered can also be used to start further operations and launch malicious attacks, such as "establishing secret session keys and credentials, smart healthcare device physical capture attack, data replaying, conducting the privileged-insider attack, malware injecting and scripting attacks, DoS attack, impersonation attack, credentials leakage and man-in-the-middle (MiTM) attacks." The malware injection attacks may be of different types, like spyware attacks, rootkit attacks, ransomware attacks, Trojan horse insertion, and launching of viruses and worms attacks. The cloud servers are considered as the trusted network entities and they performed the task of intrusion detection in a joint operation with the proxy servers, hospital servers and capture engines. The DY model does not cover all potential attacks, i.e., leakage of session states and unauthorised session key computation. Such attacks are covered by the CK-adversary model. Therefore, we have considered the CK-adversary model along with the DY model in the design of the proposed scheme.

IV. The Proposed Intrusion Detection Scheme

In this section, we provide the details of the EIDS-HS. Figure 2 illustrates the flow of execution for the various processes involved in EIDS-HS.

- The process involves registering the proxy hospital server to initiate the ad-hoc network that generates new data for fine-tuning the model, followed by the installation of the intrusion system's capture engine on the hospital's server machine. Next, an authentication process is carried out between the hospital's server and the cloud server hosting the ensemble ML Model. Meanwhile, the attacker delivers malware to intrude into the hospital's server by using phishing techniques.
- Attacker attempts to gain access to confidential data on the server or critical infrastructure by impersonating a hospital employee. Meanwhile, the ad-hoc network simulates medical servers to collect new malware samples.
- The capture engine of the proposed intrusion detection system frequently sniffs network packets from the hospital's server. At the same time, the newly collected malware is sent to the cloud server for further training to fine-tune the ensemble model.
- The capture engine uses data pre-processing techniques to clean the captured network flow.
- The cleaned network files are sent to the cloud-based server containing the ensemble machine learning model.
- The pre-processed data is analysed by the updated ensemble machine learning model.
- The ensemble machine learning model provides the prediction for the intrusion detection application, which is then sent back to the capture engine.
- Hospital authorities are alerted by the capture engine as the intrusion attempt is detected.

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

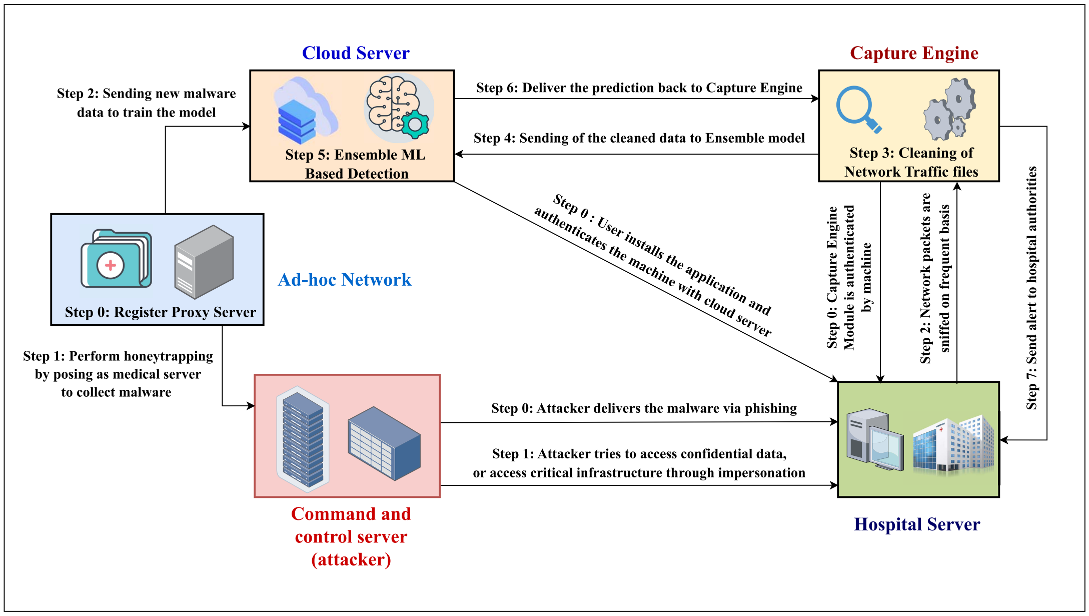

Cloud Server
Capture Engine
Step 2: Sending new malware data to train the model
Step 6: Deliver the prediction back to Capture Engine
Step 4: Sending of the cleaned data to Ensemble model
Step 5: Ensemble ML Based Detection
Step 3: Cleaning of Network Traffic files
Step 0 : User installs the application and authenticates the machine with cloud server
Step 0: Capture Engine Module is authenticated by machine
Step 2: Network packets are sniffed on frequent basis
Step 7: Send alert to hospital authorities
Step 0: Register Proxy Server
Ad-hoc Network
Step 1: Perform honeytrapping by posing as medical server to collect malware
Step 0: Attacker delivers the malware via phishing
Step 1: Attacker tries to access confidential data, or access critical infrastructure through impersonation
Command and control server (attacker)
Hospital Server

Fig. 2. Process flow diagram of EIDS-HS.

TABLE I
NOTATIONS USED IN EIDS-HS

Notation	Meaning
EIDS-HS	Ensemble-based ML-envisioned scheme to detect intrusions in the Industry 5.0-driven healthcare system
PS_i	i^{th} proxy server
HS_j	j^{th} hospital server
CS_k	k^{th} cloud server
SK_{PS_i,CS_k}	Session key of PS_i and CS_k
SK_{HS_j,CS_k}	Session key of HS_j and CS_k
$\mathcal{A}$	An adversary
AI	Artificial intelligence
ML	Machine learning (ML)
DL	Deep learning
DoS	Denial-of-service attack
IDS	Intrusion detection systems
CNN	Convolutional neural network
SVM	Support Vector Machine
KNN	k-nearest neighbors
TP	True positive
FP	False positive
TN	True negative
FN	False negative

The details of the notations used in the proposed EIDS-HS are given in Table I.

A. Dataset Acquisition and Preprocessing

We have evaluated the proposed EIDS-HS through the NSL-KDD dataset [35], an improved version of the original KDD CUP 99 Dataset [36]. The KDD dataset consists of network traffic data collected from a simulated military network that encompasses different types of network attacks, including denial-of-service attacks, probing attacks, and user-to-root attacks. In total, 5 million network connections are described by 41 features, including protocol type, service, duration, source IP address, and other network statistics. For our inclusion detection study, we employed a binary classification problem with this dataset where our objective was to classify instances as either normal or abnormal based on the provided features in the dataset. The dataset contained 79,179 normal instances and 81,161 abnormal (intrusion) instances. In order to prepare the data for our model, we performed several preprocessing steps. First, we removed any duplicate records from the dataset, resulting in a new dataset. Next, we applied standard scaling to the numeric features in the data to normalize their values. We then implemented one-hot encoding to transform the categorical features into numerical features that can be used directly by the model. Finally, Principal Component Analysis (PCA) was carried out to reduce the dimensionality of the data by extracting the most significant features while retaining the maximum amount of variance possible. The details of this process are illustrated in Algorithm 2. Algorithm 2 contains steps like the separation of the dataset into feature matrix X and target variable y. Then splitting the dataset into training and testing sets in a ratio of 70/30 was done. Moreover, there was the transformation of training and testing sets using the fitted scaler. Then one-hot encoding was applied to the categorical features in X. After that we applied label encoding for attacks in target variable y. Then fitting of the PCA transformer on the training set was done. We fitted the machine learning models on the preprocessed training set. Then there was the evaluation of the performance of the trained machine learning model on the preprocessed testing set. After that, there was the tuning of the hyperparameters of the model if it was required. Furthermore, we did the performance evaluation for all deployed machine learning models and prepare a list of intrusions as per the set criteria.

B. Machine Learning Classifiers

For our study, we employed three statistical machine-learning algorithms. Additionally, we constructed an ensemble classifier by utilizing these three algorithms.

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Ensemble Model Generation Process

```
Input: A dataset D containing labeled instances,
A parameter $k$ for KNN classifier,
A validation set $V$ of instances
Output: An ensemble classifier $C$ consisting of SVM, KNN, and DT classifiers, and their weights $w_1$, $w_2$, and $w_3$ respectively
 1: for All deployed devices do
 2:     Split the dataset $D$ into training set $T$ and testing set $E$.
 3:     for Each base model type do
 4:         Train a model $M$ using $T$.
 5:         Calculate the accuracy $a$ of $M$ on $V$.
 6:         Store the model $M$ and its accuracy $a$ in a list $L$.
 7:     end for
 8:     Normalize the accuracy values of the models in $L$ by dividing each $a$ by the sum of all accuracies.
 9:     Initialize the weights $w_1$, $w_2$, and $w_3$ to be the normalized accuracy values of SVM, KNN and DT classifiers, respectively.
10:     for Each instance in $V$ do
11:         Obtain the prediction $p_1$ of SVM, $p_2$ of KNN and $p_3$ of DT classifiers.
12:         Calculate the ensemble prediction $p = w_1 \times p_1 + w_2 \times p_2 + w_3 \times p_3$.
13:     end for
14:     Evaluate the accuracy of the ensemble classifier on $V$.
15:     if The accuracy of the Ensemble classifier is above a predetermined threshold then
16:         Return Ensemble classifier model.
17:     else
18:         Repeat from Step 2 with a different validation set $V$ until a suitable accuracy is achieved.
19:     end if
20: end for
```

Algorithm 2 Intrusion Detection Process

```
Output: List of detected intrusions
for All deployed devices do
    Load the dataset.
    Remove duplicate records from the dataset.
    Separate the dataset into feature matrix $X$ and target variable $y$.
    Split the dataset into training and testing sets in a ratio of 70/30.
    Fit the standard scaler on the training set to normalize.
    Transform the training and testing sets using the fitted scaler.
    Apply one-hot encoding to the categorical features in $X$.
    Apply label encoding for attacks in target variable $y$.
    Fit the PCA transformer on the training set.
    Transform the training and testing sets using the fitted PCA transformer.
    Fit the machine learning models on the preprocessed training set.
    Evaluate the performance of the trained machine learning model on the preprocessed testing set.
    Tune the hyperparameters of the model if it is required.
    if Performance evaluation is done for all deployed machine learning models then
        Predict the best performance.
    else
        Continue for other deployed machine learning models.
    end if
    Prepare a list of intrusions from the detection intrusions.
end for
```

One of the most often used machine learning techniques for classification and regression applications is decision trees. Based on the characteristics of the input data, a tree-like model of choices and potential outcomes is built. The branches of the tree reflect potential outcomes based on features, while the nodes in the tree represent decisions based on the features.

A well-liked supervised learning technique for classification and regression tasks is the support vector machine (SVM). SVM separates various classes by creating a hyperplane or collection of hyperplanes. An algorithm is an effective tool for classification jobs since it seeks out the hyperplane that optimises the margin between the classes.

A non-parametric approach called K-Nearest Neighbors is utilized in classification and regression problems. The algorithm finds the K data points in the training set that is closest to a new data point using a distance metric like the "Manhattan or Euclidean distance." The new data point's class is then chosen by casting a majority vote among the classes of its K closest neighbours. KNN, which can handle both category and numerical data, can handle non-linear data successfully.

To create an ensemble model of SVM, Decision Tree, and KNN using a weighted average approach, we first train each model separately on the same dataset. Then, we combine their predicted probabilities by taking a weighted average of their probabilities, where the weights are assigned based on each model's performance on a validation dataset. This weighted average approach allows us to leverage the strengths of each model and improve the overall accuracy of the ensemble model. We evaluate the performance of the ensemble model on a validation dataset. The weights of each model as needed to optimize the performance of the ensemble model are adjusted. The details of building the model are provided in Algorithm 1.

The support vector machine (SVM), KNN, and decision tree are the most suitable machine learning models for such problems, therefore, we have used them for the detection of intrusion in the proposed EIDS-HS.

In Algorithm 1, p_1, p_2 and p_3 are the weighted values of predictions of different machine learning models, i.e., SVM, KNN and decision tree (DT) with different weights (i.e., w_1, w_2 and w_3). The final value p provides more robust predictions by leveraging the strengths of different models, i.e., SVM, KNN and decision tree to overcome any weaknesses or limitations of a single model.

Remark 1 (Usefulness of Ensemble Model Generation Process): The ensemble models aggregate several different individual models to improve the system's performance. Ensemble models are less susceptible to over-fitting as each individual model in the ensemble has been trained on a different subset of the data. Also, ensemble models can reduce bias in the final predictions by combining the outputs of multiple models, thereby reducing the impact of any single model's limitations. Finally, ensemble models can provide more robust predictions by leveraging the strengths of different individual models to overcome any weaknesses or limitations of a single model. Overall, ensemble models have been shown to be effective in improving the performance of the intrusion detection process. A similar task is performed in Algorithm 1.

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

TABLE II
DETAILS OF SIMULATION PARAMETERS

Parameter	Value
Operating system	Ubuntu 18.04.5 LTS
Platform used	Google colab environment
Processor	2 X Intel(R) Xeon(R) CPU @ 2.20GHz
GPU	12 GB NVIDIA Tesla K80
Random access memory (RAM) size	12 GB
Programming language	Python 3.8
Used libraries	Scikit-learn (sklearn)
Dataset used	NSL-KDD dataset [35]
Machine learning models	support vector machine (SVM), K-Nearest Neighbours (KNN), and decision tree

V. Practical Implementation

This section outlines the practical implementation of the proposed EIDS-HS, including specific details on the hardware and software used. We utilized a processor with 2 X Intel Xeon and a clock speed of 2.20GHz, along with a random access memory (RAM) size of 12 GB. Our training environment was set up on the Google Colab platform, which used the Ubuntu 18.04.5 LTS platform for the simulation. To implement our scheme, we utilized Python 3.8 and the Scikit-learn (sklearn) library. The details of simulation parameters are given in Table II.

1) Evaluation Metrics: The proposed model was evaluated using four parameters: TP (true positive), FP (false positive), TN (true negative), and FN (false negative). TP refers to the number of times a malicious program is correctly identified as malicious, while TN refers to the number of times a benign program is correctly identified as benign. On the other hand, FP refers to the number of times a benign program is mistakenly identified as malicious, and FN refers to the number of times a malicious program is mistakenly identified as benign.

Accuracy: It is “a very important performance parameter, which is measured as all correctly identified cases”. Therefore, utilizing accuracy is imperative when classes are equally important. It can be estimated as

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}. \quad (1)$$

Recall: The number of “positive class predictions made out of all positive examples in the dataset is calculated as recall”. It is calculated as

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (2)$$

Precision: The number of “positive class predictions that actually belong to the positive class is measured by precision”. It is calculated as

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (3)$$

F1-score: Also referred to as “F1-measure, which is calculated through the harmonic mean of precision and recall”. “It provides the exact estimate of the incorrectly classified cases than the accuracy”. It is mathematically denoted as

$$\text{F1-score} = \frac{2(P \times R)}{P + R}. \quad (4)$$

where P is precision and R is recall.

TABLE III
PERFORMANCE OF IMPLEMENTED MACHINE LEARNING METHODS

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	93.68%	91.86%	92.35%	92.10%
SVM	89.75%	90.96%	88.62%	89.77%
KNN	93.50%	92.17%	94.24%	93.19%
SKD Ensemble	95.12%	93.11%	95.27%	94.18%

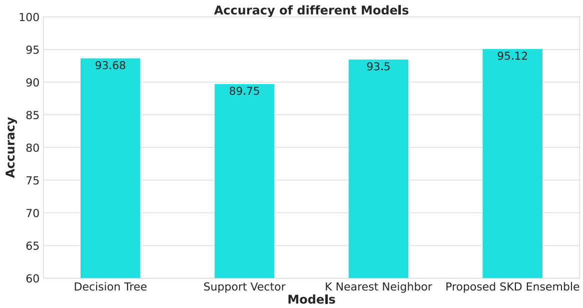


Fig. 3. Accuracy values of different methods used in proposed EIDS-HS.

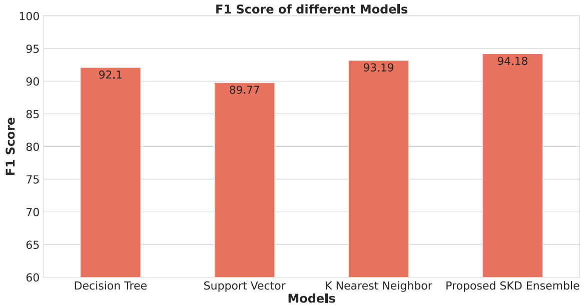


Fig. 4. F-1 scores of different methods used in proposed EIDS-HS.

Table III provides a summary of the performance metrics achieved by our model, including accuracy, recall, precision, and F1-score. These metrics are critical for evaluating the effectiveness of our model in accurately classifying instances. Figures 3 and 4 provide a graphical representation of the performance metrics, allowing for a more intuitive understanding of our model’s performance.

The performance of various implemented machine learning methods in the proposed EIDS-HS is given in Table III. From the data given in Table III, it is clear that EIDS-HS has achieved the best accuracy value in the case of the SKD ensemble method, which is approximately 95.12%.

VI. Comparative Study

In this section, we have compared the effectiveness of EIDS-HS, a proposed intrusion detection scheme, with other existing schemes. The comparisons are done in terms of accuracy, computational cost and “security and functionality features”.

A. Comparison of Accuracy Values of Different Schemes

The accuracy of the considered schemes of Immanuel Jeo Sherin and Radhika [28], Esmaeili et al. [29], Zou et al. [30], Ketepalli and Bulla [31], and the proposed EIDS-HS are

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
COMPARISON OF DIFFERENT SCHEMES

Scheme	Accuracy (in %)
Sherin *et al.* [28]	81.67
Esmaeili *et al.* [29]	82.80
Zou *et al.* [30]	85.95
Ketepalli *et al.* [31]	94.74
Proposed EIDS-HS	95.12

TABLE V
COMPARISON OF COMPUTATIONAL COMPLEXITY

Scheme	Sherin *et al.* [28]	Esmaeili *et al.* [29]	Zou *et al.* [30]	Ketepalli *et al.* [31]	Proposed EIDS-HS
Computational complexity	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$

81.67%, 82.8% (SVM), 85.95%, 94.74%, and 95.12%, respectively.

The details of the comparison between the proposed EIDS-HS and existing schemes are presented in Table IV. The results demonstrate that the computational complexity of the proposed EIDS-HS is superior to that of the existing schemes.

B. Comparison of Computational Costs of Different Schemes

This section provides a comparison of the computational complexity of the proposed EIDS-HS with existing schemes. Table V presents the findings. The computational complexity of different schemes, like, the scheme of Immanuel Jeo Sherin and Radhika [28], Esmaeili et al. [29], Zou et al. [30], Ketepalli and Bulla [31], and the EIDS-HS are $O(n)$, $O(n)$, $O(n)$, $O(n)$, and $O(n)$, respectively. The results of the comparison show that the computational complexity of all the analyzed schemes is linear. Moreover, it is evident that the proposed EIDS-HS has a comparable computational complexity to the existing schemes.

C. Comparison of Security and Functionality Features of Different Schemes

This section provides a comparison of the security and functionality features of the proposed EIDS-HS with existing schemes. Table VI presents the findings. From the information given in Table VI, it is clear that EIDS-HS provides better security and extra functionality features as compared to the other existing schemes.

TABLE VI
COMPARISON OF SECURITY AND FUNCTIONALITY FEATURES

Features	Sherin *et al.* [28]	Esmaeili *et al.* [29]	Zou *et al.* [30]	Ketepalli *et al.* [31]	Proposed EIDS-HS
ΥF_1	81.67%	82.80%	85.95%	94.74%	95.12%
ΥF_2	NO	NO	NO	NO	YES
ΥF_3	NO	NO	NO	NO	YES
ΥF_4	NO	NO	NO	NO	YES
ΥF_5	NO	NO	NO	NO	YES
ΥF_6	NO	NO	NO	NO	YES
ΥF_7	YES	YES	YES	YES	YES
ΥF_8	NO	NO	NO	NO	YES

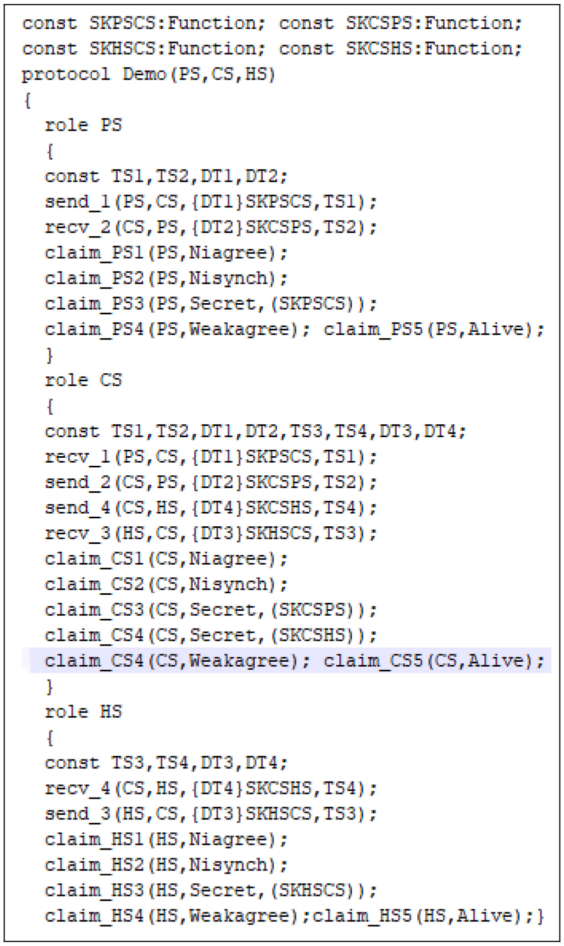
ΥF_1: “Accuracy of the scheme;”ΥF_2: “Provides secure session key establishment;” ΥF_3: “Follow standard mutual authentication among different entities;” ΥF_4: “Provides secure data exchange;” ΥF_5: “Protection against data leakage attack;” ΥF_6: “Maintains the integrity of the exchanged data;” ΥF_7: “Availability of machine learning/ deep learning-based intrusion detection process;” ΥF_8: “Availability of formal security verification using Scyther tool;” YES: “a scheme is secure or it supports a functionality feature”; NO: “a scheme is insecure or it does not support a functionality feature”.

VII. SECURITY ANALYSIS

In this section, we provide the details of the conducted security analysis of the proposed EIDS-HS. The details are given below.

- In the presented system, we have used entities like proxy servers, which do the task of honey trapping and collecting the data related to various cyber-attacks. There are also some hospital servers, which provide services to the various users of the hospital network. Moreover, we also have cloud servers, which take participation in the detection of malware attacks. There are some deployed capture engines, which are some programming modules and they reside inside the hospital servers. For the secure communication of the proxy servers and cloud servers and cloud servers and hospital servers, we need to use some authentication and key establishment protocols, which help to perform the mutual authentication among the various entities and provide the secure session key establishment among the various entities. For the authentication and key establishment any existing standard protocol (i.e., the protocol of Jia et al. [37]) can be utilized. After the execution of this protocol, there will be the secure session key establishment between the proxy server and cloud server (i.e., the session key SK_{PS_i,CS_k}) and hospital server and cloud server (i.e., the session key SK_{HS_j,CS_k}). It is also recommended to use the short-term secret values (random nonce values and timestamp values) and long terms secret values (i.e., secret keys and identities of various entities) for the creation of the session keys, due to which we get distinct keys for different devices/servers in various established sessions. Thus we get protection against the unauthorized session key computation attack in the EIDS-HS.
- Moreover, in the EIDS-HS in all exchanged messages, it is recommended to use the data only in the encrypted format not in plain text format. Apart from that in all exchanged messages, we use various random nonce values and timestamp values. Thus in the EIDS-HS, we get protection against the untraceability problem as well as get the anonymity of the exchanged messages.
- Storing sensitive information in device memory can pose a significant security risk as it can be vulnerable to attacks such as physical device capture or insider attacks. Therefore, in the EIDS-HS, it is recommended to avoid storing any secret or sensitive data in device memory to enhance the security of the system.
- Furthermore, the servers, like, the cloud servers and hospital servers store secret/sensitive data only in the secured region of their database. Under those circumstances, the attacker does not get any chance to deduce the secret/sensitive data from the memory of those servers. Therefore attacker does not get any chance to launch the attack like, the stolen verifier on the EIDS-HS.
- The task of malware detection is performed through the cloud servers, capture engine and proxy servers. When

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.



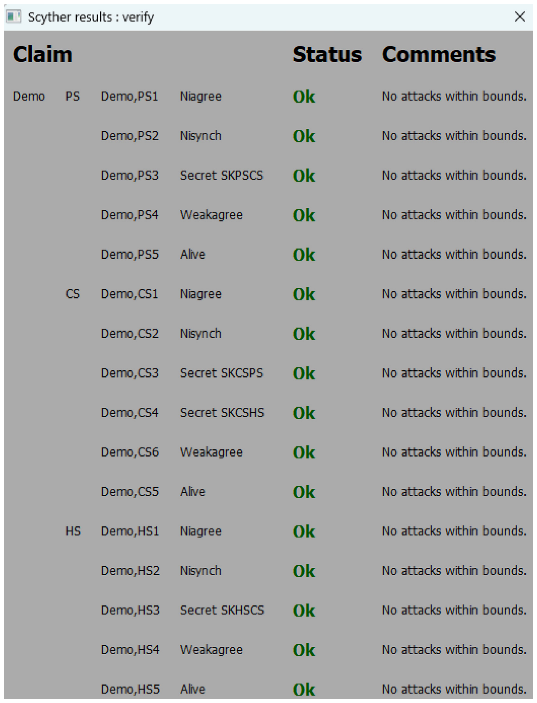
```
const SKPSCS:Function; const SKCSPS:Function;
const SKHSCS:Function; const SKCSHS:Function;
protocol Demo(PS,CS,HS)
{
  role PS
  {
  const TS1,TS2,DT1,DT2;
  send_1(PS,CS,{DT1}SKPSCS,TS1);
  recv_2(CS,PS,{DT2}SKCSPS,TS2);
  claim_PS1(PS,Niagree);
  claim_PS2(PS,Nisynch);
  claim_PS3(PS,Secret,(SKPSCS));
  claim_PS4(PS,Weakagree); claim_PS5(PS,Alive);
  }
  role CS
  {
  const TS1,TS2,DT1,DT2,TS3,TS4,DT3,DT4;
  recv_1(PS,CS,{DT1}SKPSCS,TS1);
  send_2(CS,PS,{DT2}SKCSPS,TS2);
  send_4(CS,HS,{DT4}SKCSHS,TS4);
  recv_3(HS,CS,{DT3}SKHSCS,TS3);
  claim_CS1(CS,Niagree);
  claim_CS2(CS,Nisynch);
  claim_CS3(CS,Secret,(SKCSPS));
  claim_CS4(CS,Secret,(SKCSHS));
  claim_CS4(CS,Weakagree); claim_CS5(CS,Alive);
  }
  role HS
  {
  const TS3,TS4,DT3,DT4;
  recv_4(CS,HS,{DT4}SKCSHS,TS4);
  send_3(HS,CS,{DT3}SKHSCS,TS3);
  claim_HS1(HS,Niagree);
  claim_HS2(HS,Nisynch);
  claim_HS3(HS,Secret,(SKHSCS));
  claim_HS4(HS,Weakagree);claim_HS5(HS,Alive);}
```

Fig. 5. SPDL snippet of the proposed EIDS-HS.

there is any sign of malware then the cloud server shares that information with the capture engine. After that capture engine generates alerts to the hospital servers and appropriate actions are taken, i.e., blocking of IP addresses, and port numbers, which send the malicious programs (i.e., malware attacks) to the hospital network. In this way, EIDS-HS provides protection against various possible attacks.

VIII. Formal Security Verification Using Scyther Tool

This section presents details about using the Scyther tool for formal security verification. Scyther is used to do the formal security verification of proposed EIDS-HS [38], [39]. On the basis of numerous cryptographic presumptions, Scyther offers predictions. It suggests that without the secret key, an attacker would be unable to decrypt the encrypted contents. It is very important to conduct a formal security verification of a newly designed scheme through the Scyther tool as it helps to identify any possibility of various attacks, i.e., replay attack, man-in-the-middle attack, impersonation attack, credential leakage, session key leakage, etc., on the newly designed scheme. Scyther replicates user-defined security protocols using "Security Protocol Descriptive Language (SPDL)." The Scyther tool uses the Canetti-Krawczyk (CK) model, the eCK model, and nine other adversarial models in addition to the Dolev-Yao (DY) model. The tests Scyther offers are said to verify security elements including "secrecy, authentication, synchronization, aliveness, weak agreement, and agreement." In EIDS-HS, there are some basic roles for the "authentication and key agreement phase," i.e., *PS* (for proxy server), *HS* (for hospital server) and *CS* (for cloud server). The steps of EIDS-HS are coded via SPDL. In SPDL snippet of the proposed EIDS-HS different claims are given. These are related to the secret values of different entities, for example, the shared session keys of entities, i.e., proxy server *PS*, hospital server *HS* and cloud server are *CS* (SK_{PSCS}, SK_{CSPS}, SK_{CSHS}, SK_{HSCS}). Apart from that, we have claims related to the agreement of values, synchronisation of communicating entities and aliveness of entities, which are essential claims and required to set for the proposed EIDS-HS. Scyther performs several analyses on EIDS-HS using the SPDL file as input. Fig. 5 presents the SPDL snippet of EIDS-HS. The results of the Scyther tool are shown in Fig. 6. Examining the EIDS-HS further revealed that it is protected by the aforesaid claims.



Claim				Status	Comments
Demo	PS	Demo,PS1	Niagree	Ok	No attacks within bounds.
		Demo,PS2	Nisynch	Ok	No attacks within bounds.
		Demo,PS3	Secret SKPSCS	Ok	No attacks within bounds.
		Demo,PS4	Weakagree	Ok	No attacks within bounds.
		Demo,PS5	Alive	Ok	No attacks within bounds.
	CS	Demo,CS1	Niagree	Ok	No attacks within bounds.
		Demo,CS2	Nisynch	Ok	No attacks within bounds.
		Demo,CS3	Secret SKCSPS	Ok	No attacks within bounds.
		Demo,CS4	Secret SKCSHS	Ok	No attacks within bounds.
		Demo,CS6	Weakagree	Ok	No attacks within bounds.
		Demo,CS5	Alive	Ok	No attacks within bounds.
	HS	Demo,HS1	Niagree	Ok	No attacks within bounds.
		Demo,HS2	Nisynch	Ok	No attacks within bounds.
		Demo,HS3	Secret SKHSCS	Ok	No attacks within bounds.
		Demo,HS4	Weakagree	Ok	No attacks within bounds.
		Demo,HS5	Alive	Ok	No attacks within bounds.

Fig. 6. Outcome of formal security verification.

IX. Conclusion

In this article, we demonstrated that the ensemble learning can be an effective approach to detect intrusion detection for Industry 5.0-driven healthcare system. Towards this, a novel ensemble-based machine learning-envisioned intrusion detection scheme was presented (in short EIDS-HS). The security analysis of the proposed EIDS-HS was given, which proved its security against various possible attacks. The formal security verification through the Scyther automated tool also proved that EIDS-HS is robust against various possible cyber attacks. A practical demonstration of the EIDS-HS was conducted to

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.

evaluate its impact on various performance parameters. In the comparative study, it has been observed that EIDS-HS outperformed the existing schemes.

In future, we would like to optimize the ensemble learning model and extend its applicability to different healthcare settings and attack scenarios. Thus, some future goals include integrating the model into existing intrusion detection systems, conducting real-world testing via a testbed implementation, and exploring the use of other machine learning techniques, such as deep learning, for intrusion detection in healthcare.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the associate editor for providing their valuable feedback.

REFERENCES

[1] A. Adel, "Future of industry 5.0 in society: Human-centric solutions, challenges and prospective research areas," *J. Cloud Comput.*, vol. 11, no. 1, p. 40, Sep 2022.

[2] R. Gupta, P. Bhattacharya, S. Tanwar, N. Kumar, and S. Zeadally, "GaRuDa: A blockchain-based delivery scheme using drones for healthcare 5.0 applications," *IEEE Internet Things Mag.*, vol. 4, no. 4, pp. 60–66, Dec. 2021.

[3] H. R. Chi, C. K. Wu, N.-F. Huang, K.-F. Tsang, and A. Radwan, "A survey of network automation for Industrial Internet-of-Things toward industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 2065–2077, Feb. 2023.

[4] S. P. Mohanty and F. Pescador, "Introduction consumer technologies for smart healthcare," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 1–2, Feb. 2021.

[5] R. K. Nath and H. Thapliyal, "Smart wristband-based stress detection framework for older adults with cortisol as stress biomarker," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 30–39, Feb. 2021.

[6] Z. Xu, W. Liang, K.-C. Li, J. Xu, A. Y. Zomaya, and J. Zhang, "A time-sensitive token-based anonymous authentication and dynamic group key agreement scheme for industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7118–7127, Oct. 2022.

[7] T. Jabeen, H. Ashraf, and A. Ullah, "A survey on healthcare data security in wireless body area networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 9841–9854, Oct. 2021.

[8] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020.

[9] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.

[10] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscipl. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, 2018, Art. no. e1249.

[11] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Front. Comput. Sci.*, vol. 14, pp. 241–258, Apr. 2020.

[12] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[13] G. Thamilarasu, A. Odesile, and A. Hoang, "An intrusion detection system for Internet of Medical Things," *IEEE Access*, vol. 8, pp. 181560–181576, 2020.

[14] M. M. Alani and A. I. Awad, "An intelligent two-layer intrusion detection system for the Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 683–692, Jan. 2023.

[15] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, A. K. M. N. Islam, and M. Shorfuzzaman, "Permissioned blockchain and deep learning for secure and efficient data sharing in industrial healthcare systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8065–8073, Nov. 2022.

[16] Z. Chen, J. Duan, L. Kang, and G. Qiu, "Class-imbalanced deep learning via a class-balanced ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5626–5640, Oct. 2022.

[17] V. Stephanie, I. Khalil, M. S. Rahman, and M. Atiquzzaman, "Privacy-preserving ensemble infused enhanced deep neural network framework for edge cloud convergence," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 3763–3773, Mar. 2023.

[18] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

[19] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," *IEEE Access*, vol. 9, pp. 7550–7563, 2021.

[20] J. Liu, Z. Tian, R. Zheng, and L. Liu, "A distance-based method for building an encrypted malware traffic identification framework," *IEEE Access*, vol. 7, pp. 100014–100028, 2019.

[21] Y.-L. Wan, J.-C. Chang, R.-J. Chen, and S.-J. Wang, "Feature-selection-based ransomware detection with machine learning of data analysis," in *Proc. 3rd Int. Conf. Comput. Commun. Syst. (ICCCS)*, 2018, pp. 85–88.

[22] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021.

[23] P. Prasse, L. Machlica, T. Pevný, J. Havelka, and T. Scheffer, "Malware detection by analysing network traffic with neural networks," in *Proc. IEEE Security Privacy Workshops (S&P)*, 2017, pp. 205–210.

[24] M. Piskozub, R. Spolaor, and I. Martinovic, "MalAlert: Detecting malware in large-scale network traffic using statistical features," *SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 151–154, 2019.

[25] R. Abedin and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, pp. 1–22, 2022.

[26] R. Chiramdasu, G. Srivastava, S. Bhattacharya, P. K. Reddy, and T. Reddy Gadekallu, "Malicious URL Detection using Logistic Regression," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst. (COINS)*, Barcelona, Spain, 2021, pp. 1–6.

[27] R. M. Swarna Priya et al., "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Comput. Commun.*, vol. 160, pp. 139–149, Jul. 2020.

[28] V. J. Immanuel Jeo Sherin and N. Radhika, "Stacked ensemble-IDS using NSL-KDD dataset," *J. Pharmaceut. Negative Results*, vol. 13, no. 3, pp. 351–356, 2022.

[29] M. Esmaeili, S. H. Goki, B. H. K. Masjidi, M. Sameh, H. Gharagozlou, and A. S. Mohammed, "ML-DDoSnet: IoT intrusion detection based on denial-of-service attacks using machine learning methods and NSL-KDD," *Wireless Commun. Mobile Comput.*, vol. 2022, Aug. 2022, Art. no. 8481452.

[30] L. Zou, X. Luo, Y. Zhang, X. Yang, and X. Wang, "HC-DTTSVM: A network intrusion detection method based on decision tree twin support vector machine and hierarchical clustering," *IEEE Access*, vol. 11, pp. 21404–21416, 2023.

[31] G. Ketepalli and P. Bulla, "Feature extraction using LSTM autoencoder in network intrusion detection system," in *Proc. 7th Int. Conf. Commun. Electron. Syst. (ICCES)*, Coimbatore, India, 2022, pp. 744–749.

[32] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[33] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[34] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[35] "NSL-KDD dataset." Accessed: Mar. 2023. [Online]. Available: http://nsl.cs.unb.ca/nsl-kdd/

[36] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, Ottawa, ON, Canada, 2009, pp. 1–6.

[37] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Netw.*, vol. 25, no. 8, pp. 4737–4750, Nov. 2019.

[38] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of Drone environment," *IEEE Access*, vol. 8, pp. 155645–155659, 2020.

[39] C. J. F. Cremers. "Scyther: Semantics and verification of security protocols." 2022. Accessed: Nov. 2022. https://pure.tue.nl/ws/files/2425555/200612074.pdf

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 01,2024 at 15:24:02 UTC from IEEE Xplore. Restrictions apply.