Discover Applied Sciences

Research

Development of lightweight intrusion model in Industrial Internet of Things using deep learning technique

Raj Sinha[1] · Padmanabh Thakur[2] · Sandeep Gupta[2] · Anand Shukla[3]

Received: 1 March 2024 / Accepted: 19 June 2024
Published online: 24 June 2024

Abstract
Nowadays, the IDS is being used in conjunction with the IIoT system to reduce the security risk, but on the other hand, the false rate of the IDS is very high. Therefore, in this work, a pre-training method, making use of both a deep neural network and a deep auto-encoder, has been proposed for the quick prediction of assaults with increased accuracy and a reduced false rate. The replicas were expanded using hyperparameter optimization (HPO) techniques. The proposed model delivers an alternative to deep learning construction replicas through an HPO procedure incorporating the Archimedes optimization algorithm. This optimization technique can be used to determine the hyperparameter value and the ideal categorical hyperparameter combination for improved detection performance. The DS2OS dataset is used alongside numerous other indicators to evaluate the efficacy of the developed model. The various existing techniques of assault detection have also been considered to show the effectiveness of the proposed model. Through the comparative evaluation of the outcomes, it is shown that the developed model provides better performance than the other existing models. Eventually, it is discovered that the suggested security paradigm is successful in fending off a variety of internal and external threats.

Highlights

1. A hybridization of DNN and DAE is considered for accurate assaults detection.
2. Archimedes optimization algorithm has been used to improv the detection performance.
3. The percentage accuracy and precision of the developed model are 96.57 and 98.57.
4. The percentage recall and F-measure of the developed hybrid model are 98.69 and 98.35.

Keywords Industrial Internet of Things · Intrusion detection organization · Archimedes optimization algorithm · Cyber-attacks

Abbreviations

APT	Advanced persistent threat
IIoT	Industrial Internet of Things
IDS	Intrusion detection system

✉ Anand Shukla, anandshuklaR@wollegauniversity.edu.et; Raj Sinha, rajsinha2310@gmail.com; Padmanabh Thakur, tonu_arth@rediffmail.com; Sandeep Gupta, jecsandeep@gmail.com | [1]School of Computer Applications, Lovely Professional University, Phagwara, Punjab, India. [2]Department of Electrical Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand 248002, India. [3]Wollega University, Nekemte, Ethiopia.

DAE	Deep auto encoder
DNN	Deep neural network
HPO	Hyperparameter optimization techniques
FPRs	False positive rates
DL	Deep learning
SET	Sparse evolutionary training
ML	Machine learning
RLU	Rectified linear unit
AE	Auto encoder
F-M	F-measure
PR)	Precision
RC	Recall
TP	True positive
FP	False positive
FN	False negative
TN	True negative
AUC	Area under the curve
HDRaNN	Hybrid neural network
DoS	Denial of service

1 Introduction

Nowadays, the protection of critical infrastructure has become a pressing concern because of the increasing number of IoT devices and their proliferation by several industries [1]. Malware, taking advantage of zero-day vulnerabilities, is a major problem in IIoT settings. The attackers, across the globe, firstly infect the critical devices and then use techniques, such as APT, DoS, and Distributed DoS to exert control over and alter their operations [2, 3]. For the Iranian nuclear program, in 2013, hackers from Iran broke into the control system. Malware caused power outages for at least 80,000 customers in Ukraine, and most recently, SFG malware attacked European energy companies [4, 5]. These attacks have shown that the typical cyber-security methods, including security rules, authentication, firewalls, and IDSs, are still unable to ensure the safety of critical infrastructure [6]. Typically, deployment of the IDS is considered an auxiliary defense after the installation of a firewall, anti-virus software, and access control systems for the purpose of detecting assaults on IIoT devices [7, 8]. Here, the term "intrusion detection system (IDS)" refers to a software and/or hardware technique used to monitor and identify suspicious activity across networked systems [9–11].

The purpose of the paper work [10] is to offer a feature selection approach for intrusion detection systems (IDS) in the Internet of Things (IoT) that makes use of Information Gain (IG) and Gain Ratio (GR) using the top fifty percent rated features. The research study aims to provide people interested in learning more about networking data collection and analysis mining for social networks a quick and basic review of the fundamentals of the field [11]. While the former can identify known assaults but at the expense of a large number of mistakes, the latter can detect both known and novel attacks [12]. If the approach of an anomaly-based IDS can effectively identify both known and undiscovered threats that seek to infiltrate IIoTs, it might be a strong tool [13, 14].

Classical data mining methods, rules-based models, artificial intelligence approaches, and statistical models have already been reported in research for the construction of IDSs. However, owing to the overlap between normal and abnormal data, these techniques often provide high false positive rates (FPRs) [15]. Nearly all current IDS rely on outdated machine learning methods to create detection models [16]. So, there are still several obstacles that traditional machine-learning-based solutions must overcome [17]. Recently, the use of deep learning (DL) methods in IDSs has been strongly recommended in research to improve the performance of IDSs. In foreseeing the future of the IoT, DL speeds up the analysis between real-time data streams and their faster virtual counterparts [18]. Due to the superior accuracy and ease of information extraction in DL, it has supplanted more conventional forms of education [19]. This has led to a number of studies focusing on employing deep learning methods to provide new solutions to irregularities and malware detection; nevertheless, the findings are still unconvincing [20].

The majority of IDS resolutions are also extensions of technologies that are already present in other kinds of networks, including ad hoc networks for computers and mobile devices. However, the IDS, developed for such systems, is not

applicable to IoT applications because of the exceptional features of IoT-based systems, such as access to worldwide internet resources [21]. The purpose of this study is to provide a new DL-based technique for IIoT attack prediction. Using one of the most recent datasets, the suggested method identifies IIoT threats with high accuracy and reduced prediction time. Multiple performance metrics with variable bounds are defined and used to assess the proposed algorithm's efficacy. We also compare the simulation results with those of other state-of-the-art machine learning classifiers to demonstrate the effectiveness of the proposed method.

2 State of art

The DL based model, using sparse evolutionary training (SET), has been developed in the paper [22] for the analysis and detection of the most common types of threats. The SET-based prediction model has a mean accuracy of 99% with an average testing duration of 2.29 ms. The anticipated model has been determined to enhance accuracy by an average of 6.25 percent. Although this study's proposed approach demonstrates strong overall detection performance and efficiency, it's important to acknowledge certain limitations. Initially, it is important to note that the novel sparse evolutionary training (SET) based prediction model may not completely restore the optimal detection state of the original model during training. Additionally, it leads to an increase in the number of intrusion detection steps.

Guezzaz et al. [23] present a hybrid Intrusion Detection System (IDS) for Industrial Internet of Things (IIoT) security that is built on edge computing and utilizes machine learning (ML) approaches. This study utilizes a combination of K-Nearest Neighbor (K-NN) and Principal Component Analysis (PCA) to identify outliers and wrongdoers. Here, the K-NN classifier has been incorporated to enhance the detection accuracy and make relevant judgements, and the principal component analysis is used to improve feature engineering and training. The data obtained shows that, the edge based IIoT security hybrid IDS, outperforms the other recent models. This method obtains an accuracy of 99.10%, a detection rate of 98.4%, and a false alarm rate (FAR) of 2.7% on the NSL-KDD dataset, and an accuracy of 98.2%, a detection rate of 97.6%, and a FAR of 2.9% on the Bot-IoT dataset. But the present work fails to enhance the PK-IDS framework by incorporating advanced artificial intelligence techniques, while also neglecting to consider the unique characteristics of edge-based IIoT. In comparison, this paper exclusively utilizes the K-NN classifier method.

Furthermore, a DL procedure, namely a Classifier-Convolution Neural Network Memory [24], and a rule-based feature-selection algorithm [25] have also been used to provide a unique model for recognizing different kinds of attacks. Applying deep neural networks (DNN) with bidirectional LSTM as a hybrid approach in real-time poses challenges in terms of accuracy [24]. This is due to the lack of validation on dynamic datasets and the requirement for more reliable results when using various kinds or mixes of deep learning techniques on diverse datasets. In [25], two datasets, namely NSL-KDD and UNSW-NB15, have been considered to check the effectiveness of the DL based model. It is shown that the DL based approach has a 99.0% accuracy rate, a 99.0% detection rate, and a 1.0% false positive rate (FPR) for the NSL-KDD dataset, and a 98.9% accuracy rate, a 99.9% detection rate, and a 1.1% FPR for the UNSW-NB15 dataset. But, by incorporating mixed rule-based characteristic collection, the suggested method achieves enhanced consistency by selectively utilizing relevant features for class classification in the data sets. This model may not be suitable for all features.

Moreover, hybrid neural networks (HDRaNN) have been addressed for the detection of attacks in the IIoT network [26]. DS2OS and UNSW-NB15, two datasets relevant to IIOT security, are used to assess the suggested approach. However, there are a lot of security and privacy issues with the integration of physical and cyber systems. The suggested model's consistency is enhanced by the hybrid rule-based feature collection approach, which uses a restricted number of suitable features for class categorization inside the data sets. This model does not support all functionality equally.

As a hybrid model, the HDRaNN combines the strengths of a deep random neural network with a multilayer perceptron using dropout regularization. This technique uses two IIoT security-related datasets, DS2OS and UNSW-NB15. The performance of the proposed method is measured in a variety of ways. Some of them are accuracy, precision, recall, F1 score, log loss, Area Under the Curve (AUC), and Region of Convergence (ROC). For DS2OS and UNSW-NB15, the HDRaNN obtained an accuracy of 98% and 99%, respectively, while categorizing 16 various kinds of cyberattacks. Comparing the proposed strategy's performance metrics against those of other cutting-edge attack detection approaches allows for an accurate assessment of its effectiveness. HDRaNN performed better than other DL-based approaches, according to the results obtained. To prevent biased categorization caused by over-fitting and under-fitting, this research employs the synthetic minority over-sampling (SMOTE) method [27]. But not every cyberattack can be addressed by the approach this study proposes [27]. Almaiah et al. [28], presents a DL based system with two-stages of cyber security and privacy. But this leads to the presentation of the two suggested methods in this work whenever a large data flow is required

to get the desired results. To attain the necessary degree of anonymity and security, a blockchain system is first built in which all participating units are recorded, validated, and eventually certified utilizing smart (BiLSTM). The publicly available IoT-Botnet and ToN-IoT datasets provide the foundation for the experimental results. Simulation findings compared to those of benchmark models verify the proposed framework's higher performance. Using the benefits of the IoT for industrial process management, the IIoT is a groundbreaking endeavor to establish a smart manufacturing eco-system. The following sectors and services benefit greatly from IIoT's fast expansion:

1. IoT devices are utilized to monitor, sense, and track equipment, patients, and medications in healthcare systems.
2. Internet of Things (IoT) devices are used for farm monitoring, smart watering of plants, and inventory management in the agricultural business.
3. Supply chain businesses can't function without the transportation and logistics sectors. (IoT) devices are utilized to pinpoint the exact position of a moving vehicle in this context.

It is also used in calculating the product's expected delivery date. Using the IIoT, the energy industry can keep track of the grid, its invoicing, and its leakage monitoring. IoT devices are utilized to manage warning systems, sense crisis signals, track underground miner activity, and monitor shipments in the mining sector. ICS, which encompasses things like SCADA networks and PLCs, is a term used to describe the robust automation sector (PLC). The majority of cyberattacks, like Stuxnet, the German assault, the Shamoon attack, Mirai, etc., target industrial automated systems.

The study illustrates the learning paradigms that are being applied in industry, as well as the architecture, security, and privacy concerns that are being faced. Furthermore, the report delves into a multitude of research issues that are intended to facilitate future rectifications in the methods that are utilized to address odd complications that arise in the sectors [29]. An anomaly detection method for Internet Intrusion Control Systems (IICSs) is proposed in this research [30]. Despite being designed to guard against risks based on cyberattacks, Network Intrusion Detection Systems (NIDSs) have a challenging problem when it comes to gathering data that will be used to create an intelligent NIDS that can effectively identify both new and ongoing assaults. The study presents a unique method for intrusion detection on the Internet of Things (IoT), which is accomplished via the use of a specialized deep learning algorithm [31]. However, these methods are limited to identifying attempted network intrusions.

The protection of such devices is now a significant concern because of the fast increase in the number of applications and devices that are connected to the Industrial Internet of Things (IIoT). Industrial firms are a common target of cyberattacks. IoT devices provide cybercriminals with numerous entry points into the industrial process. The classic model incorporates robust safeguards into the network infrastructure. Industrial control systems need a reliable intrusion detection technique to stave off threats.

3 IoT concept at industrial platform

Intelligent sensors, actuators, and other devices, such as radio frequency identification tags, are used in the Industrial Internet of Things (IIoT) to improve industrial and manufacturing processes. These gadgets operate together as a network to gather, share, and analyze data. Increased dependability and efficiency are facilitated by the process's insights. IIoT, sometimes referred to as the industrial internet, is used by several sectors, including manufacturing, energy management, utilities, and the oil and gas sector.

IIoT leverages the data generated by dumb machines that have been in industrial settings for years by using real-time analytics and smart machine capabilities. The underlying principle of IIoT is that intelligent robots are superior to people not only in real-time data capture and analysis but also in conveying critical information that may expedite and improve the accuracy of business choices. IIoT especially has the potential to improve supply chain efficiency overall, traceability of the supply chain, sustainable and green practices, and quality control in the industrial sector.

A system of interconnected intelligent devices known as the Internet of Things (IIoT) is used to monitor, gather, share, and analyze data. The following components make up each industrial IoT ecosystem:

- Connected devices have the ability to perceive, transmit, and retain data about their own state.
- Infrastructure for the transmission and exchange of data, both in the public and private domains.
- Data analytics and software that transform raw data into valuable business insights.

- Storage for the data produced by the Industrial Internet of Things (IIoT) devices.
- Consumers.

In order to provide consumers with services that are enabled by intelligence, the architecture of the IIoT has been comprised of computational objects that are coupled with the infrastructure of the IoT. As an additional point of interest, the Internet of Things network has been designed as a four-layer architecture, which includes the device layer, the network layer, the infrastructure layer, and the application layer. The Fig. 1 is a graphic representation of the taxonomy basic architecture diagram of the Internet of Things (IoT).

3.1 Typical IIoT attack categories

Industrial Internet of Things (IIoT) systems are open to a variety of threats that might jeopardize their security, interfere with their regular operations, or endanger vital infrastructure. Creating successful cybersecurity tactics requires an understanding of these prevalent IIoT threat types. Let's examine a few of the most common attack methods seen in the IIoT environment:

1. *Denial-of-Service (DoS) attacks* This is a kind of cyberattack where the attacker's goal is to stop a computer or other equipment from working normally so that the intended users cannot use it.
2. *Man-in-the-Middle (MitM) attacks* One frequent kind of cybersecurity attack that lets attackers listen in on two targets' conversations is the MitM attack. Attackers may use devices or network infrastructure flaws as openings to intercept, alter, or introduce malicious instructions.
3. *Device exploitation* Hackers seek to acquire unauthorized control or access by exploiting vulnerabilities that are present in equipment connected to the Industrial Internet of Things (IIs). They are able to undermine the operation of the device and possibly seize control over the whole IIoT system by preying on security flaws such as default or weak passwords, old firmware, or software that has not been patched.
4. *Physical attacks* Physical assaults include the deliberate manipulation of IIoT devices or infrastructure components. Adversaries have the potential to physically breach the devices in order to manipulate sensors, introduce harmful code, or interfere with the functioning of vital equipment. Physical assaults provide a substantial hazard to the reliability and security of industrial operations.
5. *Data interception and tampering* Devices, networks, and cloud platforms must be able to share data in a smooth manner in order for IIoT systems to function properly. The purpose of this kind of assault is to affect the meaning or accuracy of the data that is being attacked.
6. *Supply chain attacks* For the purpose of infiltrating a target's system or network, a supply chain assault makes use of tools or services provided by a third party. Master the art of preventing assaults on supply chains. It is possible for them to acquire unauthorized access to or control over the IIoT system by inserting malicious code or interfering with the devices. This poses enormous hazards to the whole infrastructure.

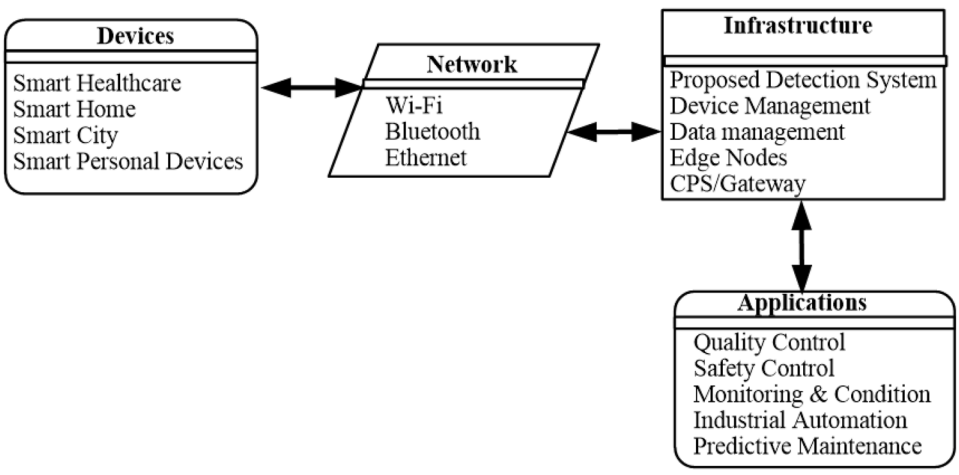

Fig. 1 A fundamental overview of the creation of the architecture for the IIoT

7. *Firmware and software vulnerabilities* For optimal functionality, IIoT devices often depend on both software and firmware. Those who want to obtain unauthorized access, influence the operation of the device, or introduce malicious malware into the system may take advantage of vulnerabilities that exist within the firmware or software.

Organizations should take preventative actions to strengthen the security of their IIoT systems by learning about these typical IIoT threats. To reduce the risks associated with these attack vectors, it is vital to implement strong security controls, update software and firmware periodically, perform vulnerability assessments, and use safe coding and configuration standards.

4 Proposed system

The proposed model consists of a deep autoencoder (DAE) combined with a deep neural network (DNN). The model is developed using hyperparameter optimization (HPO) procedures. This research provides an alternative solution to deep learning structure models through a HPO process that has the Archimedes optimization algorithm (AOA). Therefore, Fig. 2 presents the exact workflow of the model that is used to detect attacks from the dataset.

The steps, used in Fig. 2, is briefly explained in the followed sub-section.

4.1 Description of dataset

The free dataset, namely DS2OS, often used to test the accuracy and effectiveness-based cybersecurity programs, is available in [32]. It discusses several attacks, like smart homes, smart factories, smart buildings, etc. as they pertain to sensors and apps. There are a total of 35,952 samples and 13 characteristics available in this dataset. There are 34,793,575 regular data values and 100,017 outlier data values, organized into 8 categories. Also, 2,500 missing values for the "Value" feature, and 148 missing values for the "Accessed Node Type" feature are available in this dataset. The description of the assault's values, existing in the dataset, is given in Table 1.

4.2 Dataset pre-processing

The preparation of data, in line with the understanding capability of ML/DL method, is considered as the first phase of accurate data examination. Two feature fields, "Accessed Node Type" and "Value" are lacking data in this collection. There are 148 "NaN" entries in the "Accessed Node Type" data column. There is a high risk of losing useful information if these 148 rows are deleted since this functionality makes use of category data. For this reason, the "NaN" value has been replaced with the malicious "Malicious" value. There are also blanks in the "Value" column. In their place are some more anticipated values with actual significance. Values of 1.0, 0.0, and 20.0 stand "True", "False", and "Twenty" correspondingly. Therefore, the data in this collection includes both numerical and categorized information. All numerical information may be broken down further into two more distinct types: continuous and discrete. The values of a categorical variable might be either ordinal or nominal. All columns in the dataset, excluding "Value" and "Timestamp" include category nominal variables. Both columns include numeric constants. Categorical information must then be transformed into feature vectors. Label encoding is used here to transform the category encoding allows for a constant number of features while still being easily integrated into DL methods and requiring less processing time than one-hot encoding.

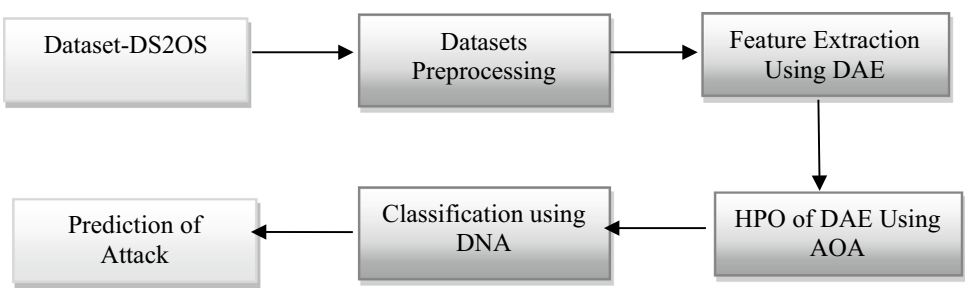

Fig. 2 Steps used in the proposed model

Table 1 Descriptions of the occurrence of the assault's values

Attacks	Description
Denial of service	When a user has problems connecting to or using a computer, printer, or other networked device, this is known as a (DoS) attack. Several systems, including email and others, may be vulnerable to this assault. The attacker floods the network with ambiguous traffic in this kind of assault. This procedure causes the network to become overloaded, which in turn causes services to be unavailable to end consumers
Malicious control	Attackers in MC can snoop on network traffic by exploiting flaws in the target application. Application backdoor attacks are used to gain remote access to a computer. The backdoor may be intentionally installed to gain access to private company or customer data
Data type probing	When a rogue node creates an undesired data type, this is known as data corruption (DP). Servers' inner workings may be probed using this technique. To create a topological map of the target network, these attacks collect and analyze data
Scan	To discover an open port and feat a known susceptibility in the targeted service, this attack technique to a wide variety of server port numbers on a host. The goal is to test a large number of potential victims and keep track of the ones who respond favorably or are valuable to the attacker
Malicious operation	Malware is often to blame when MO occurs. It's anything that may divert attention away from the primary task at hand in the system. This assault severely degrades the functionality of the equipment
Wrong setup	When an attacker compromises WS, they may get access to sensitive system data
Spying	Attackers engage in espionage when they discover and exploit security flaws in a target system in order to get unauthorized access to the target's data through a hidden backdoor. When an attacker tampers with the data, it poses a serious risk to the whole system
Normal	Data is considered normal if and only if it is without error

4.3 Data classification

Specifically, the created deep organized prototypical is a hybrid perfect that employs both a deep auto encoder (DAE) for pre-training and a detection procedure. Firstly, the data must be cleaned and pre-processed. Then, the deep model, using HPO, has been further refined to find one that yields optimal detection results. At this point, sufficient data has been collected to evaluate classifiers and determine which one yields the optimal model. There is a substantial effect of the hyper-parameter settings of the DL algorithm. The tuning hyperparameters were processed by the HPO method. Pre-training, using DAE, has been employed in the proposed IDS network for feature extraction and fine-tuning with DNN construction. In a DAE, encoding and decoding are used in tandem with feature extraction. The feature representation retrieved from the dataset is a representation of the bottleneck layer with a lower feature. The fine-tuning method for a DNN involves transferring the outcome of an encoding structure, together with values.

The DAE architecture uses the pre-processed dataset (*X*) as its input layer. The results, from the DAEs, are consistent with *X* data, or *X*-like data. The decoding layer's structure and data are not passed on to the DNN model. Data is classified into binary classes and then further categorized into multiclasses using the DNN model's training. The optimal model is obtained by hyper-parameter adjustment by monitoring the attack classification detection rate.

4.3.1 Deep auto-encoder model

The major focus of this study, in part, is to develop a reliable IDS by improving the representation of low-dimensional characteristics via their extraction from raw data. The purpose of the feature extraction procedure is to improve the DS2OS dataset's ability to identify and classify binary (normal or anomalous) attacks. Multiple hidden layers characterize a DAE, which is an AE layers of a DAE enable the AE to mathematically understand more complicated data designs. The encoding procedure on a simple AE with one hidden layer involves a mapping from input layers to the hidden layers. Decoding more encoder-decoder pairings may be found in a DAE with more hidden layers. In all, there are five discrete levels to the DAE architecture. The first stage of DAE is when encoder E1 encodes input X, encoder E2 encodes encoder E1's output, and encoder E3 encodes encoder E2's output. The expression, $Z = E3(E2(E1(X)))$, describes the intermediate layer encoding process. For an AE vector encoder 'h' in a layer, we get $h = f(W \cdot X + b)$, where W is a weight vector, X & b are vector and bias. In forward propagation, the buried layer l vector encoding purpose develops Eq. (1)

$$h^{(l+1)} = f\left(W^{(l)} \cdot h^{(l)} + b^{(l)}\right) \tag{1}$$

So, each layer can be written as $E1 = f\left(W^{(1)} \cdot X + b^{(1)}\right); E2 = f\left(W^{(2)}\right) \cdot E1 + b^{(2)}$; and $Z = E3 = f\left(W^{(3)}\right) \cdot E2 + b^{(3)}$. The order of operations during decoding is reversed from that of encoding, with the first decoder being processed last. In the last step of reconstruction, (X) = D1(D2(D3(E3(E2(E1(X)))))). Since DAE uses a decode function for layer X = f(WT·h + b′) looks like this (X) = D1. The neurons in a neural network are activated or deactivated using a mathematical operation called the activation function (.) applied to the corresponding output signal. For example, if we want the output value to be somewhere between 0 and 1 or 1 and 1, then we may use an activation function to map that range.

The DAE uses the distance function among the original (X) and the rebuilt (X) as its cost function. Mean squared error loss is used to determine the cost or loss of an activation function is given in (2):

$$J\left(w, b, x^i, \hat{x}^i\right) = \frac{1}{2}x^i - \hat{x}^{i2} \tag{2}$$

After the input data is normalized to a range from 0 to 1, a nonlinear sigmoid function may be used to rebuild the output layer. For function, an input may be any binary integer or input with a range between 0 and 1. With respect to m-data training as a whole,$J(w, b) = \frac{1}{m}\sum_{i=1}^{m} J\left(w, b, x^i, \hat{x}^i\right)$ is given as:

$$J(w, b) = \frac{1}{m}\sum_{i=1}^{m}\left[x^i \log\left(\hat{x}^i\right) + \left(1 - x^i\right)\log\left(1 - \hat{x}^i\right)\right] \tag{3}$$

To do this, backpropagation periodically adjusts of every node in every layer. Almost little value is lost at the best possible price. The information on the layer that occurs after the AE training phase is complete. Transfer learning (TL) entails feeding the encoded structure (Z) into a DNN classifier. With the help of TL, we can easily transmit the AE's weight and bias values to our classifier while still preserving the Z-encoding structure of the AE.

In order to train and learn the classifier model, a pre-training phase is performed using the AE feature extraction approach. With the use of transfer learning, HPO accompanied by an AE may alter the classifier model's education. Improving the IDS calls for tweaking the AE model's hyperparameters as well. The activation function, the learning rate, the function loss was all tuned as part of the hyperparameter tuning procedure for the model feature extraction using AE. The AE metrics were determined by tracking the AE model's loss value.

4.3.2 Deep neural network model

As a classifier, a DNN algorithm was implemented into a model for detecting attacks. Automatic leads to the encoding process. DNN architecture is fed input data X in the form of training output Z produced by the AE process. To the DNN's output y layer, another is appended [33]. The AE's weight and bias values are then used as pretraining values in a retraining procedure to acquire knowledge of the output y. y, the result, may be expressed as:

$$\hat{y} = f\left(W^{(l)} \cdot h^{(l)} + b^{(l)}\right) = f\left(z^{(l+1)}\right) \tag{4}$$

Here, $l + 1$ is the depth of the topmost layer and f(.) is the activation function of the AE structural function. The rectified linear unit (RLU) function provides benefits in comparison to other activation functions. For the hidden layer, we tested various kinds of RLU activation functions. We are employed in the output layer. Before the DNN could be trained, an initial parameter was set. The DAE encoding technique yielded the values required to be set to a tiny random number (say, a distribution centered on zero, such as n(0,0.1)) to begin with. Due to rounding, the final result, y, is rather near to the correct value. Difference desired value yi is calculated at each output node. The error value in the hidden unit is intended by averaging the relative importance of the error nodes that take h I l as inputs. Both the binary cross-entropy and the sigmoid functions are used. Figure 3 presents the architecture of the model.

4.3.3 Archimedes optimization algorithm (AOA)

The AOA is a procedure that uses data from a population [34, 35]. The proposed method uses DAE for HPO, where the population members themselves serve as the objects and parameters submerged in the environment. To the same extent as it is used in other population-based metaheuristic algorithms, acceleration is utilized in AOA. At this point in time, the

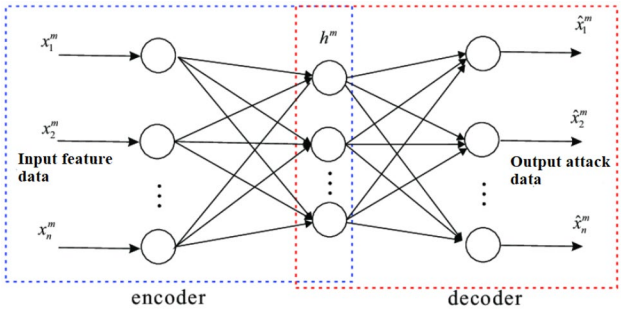

Fig. 3 Architecture Diagram of DNN

initial fluid location of each individual item is also first established. After an initial population's fitness has been assessed, AOA will continue to iterate until a termination condition is met. The density and volume of all objects are revised by AOA at each cycle. Once an item collides with another nearby object, its acceleration is adjusted accordingly. The new location of an item is calculated using its current as follows:

a. *Algorithmic phases* It is possible to see the AOA as a global optimisation approach due to its theoretical incorporation of both exploration and exploitation. The suggested AOA's mathematical steps are as follows.

This subsection presents the AOA algorithm's mathematical preparation. It is possible to consider AOA to be a global optimization approach due to the fact that it potentially combines both exploration and exploitation procedures. The following are the mathematical stages of the proposed AOA.

Step 1: Initialization put everything in its default place using (5) as a starting point:

$$O_i = lb_i + rand \times \left(ub_i - lb_i\right); i = 1, 2, \ldots, N \tag{5}$$

For every population of N objects, O_i represents the ith item. The search-space is bounded below by lb_i and above by ub_i.

Set the volume of the ith object to (6):

$$\begin{aligned} den_i &= rand \\ vol_i &= rand \end{aligned} \tag{6}$$

Here *rand* represents a vector that creates a random value between zero and one.

Finally, set the ith object's acceleration to zero, using (7):

$$acc_i = lb_i + rand \times \left(ub_i - lb_i\right) \tag{7}$$

This step involves conducting a fitness analysis on the initial population and picking the fittest item. Assign x_{best}, den_{best}, vol_{best}, and acc_{best}.

Step 2: Revision of the thickness and volume. For repetition t+1, object i's density and volume are modified using (8):

$$\begin{aligned} den_i^{t+1} &= den_i^t + rand \times \left(den_{best} - den_i^t\right) \\ vol_i^{t+1} &= vol_i^t + rand \times \left(vol_{best} - vol_i^t\right) \end{aligned} \tag{8}$$

Here, vol_{best} and den_{best} represent the data volume and rand is a chance integer drawn from an unchanging distribution. There is initial chaos when things collide, followed by an attempt at balance. The AOA does this with the aid of the transfer operator 'TF', which changes the focus of the search from exploration to abuse, as indicated by (9).

$$TF = exp\left(\frac{t - t_{max}}{t_{max}}\right) \tag{9}$$

The time, it takes to transmit a single unit of TF, progressively grows until it achieves the unit value. In (9), t represents total numbers of iterations. The density-reducing factor, 'd', helps AOA with its global-to-local search. Using the number (10), it goes down over time:

$$d^{t+1} = exp\left(\frac{t_{max} - t}{t_{max}}\right) - \left(\frac{t}{t_{max}}\right) \tag{10}$$

The value of d^{t+1} diminishes with time, allowing convergence to a previously defined sweet spot. It is important to keep in mind that this variable must be managed in such a way that exploration and exploitation are kept in a healthy balance in AOA. In the event of a collision (TF of 0.5), a (mr) is chosen, and the acceleration of the object is modified for the next iteration (t + 1) by the formula (11):

$$acc_i^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_i^{t+1} \times vol_i^{t+1}} \tag{11}$$

Here, den_i vol_i, and acc_i are the respective values for item '*i*' whereas acc_{mr}, den_{mr}, and vol_{mr} represent random substance, respectively. It's worth noting that a TF of 0.5 guarantees exploration occurs on one-third of iterations. The exploration–exploitation dynamic is modified when a value other than 0.5 is applied. Phase of exploitation, or *Step 2*, object acceleration is updated each iteration t + 1 using (12) if TF > 0.5, indicating no collision among objects.

$$acc_i^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i^{t+1} \times vol_i^{t+1}} \tag{12}$$

where acc_{best} is the hastening of the greatest object.

Step 3: Adjusting acceleration, using the formula (13), to yields the percentage change:

$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} - \min(acc)}{\max(acc) - \min(acc)} + l \tag{13}$$

where *u* and *l* are the normalization range and are defined as 0.9 and 0.1. Every agent's proportion of step change is calculated using the $acc^{t+1}{}_{i\text{-}norm}$. A high acceleration value designates that object '*i*' is in the exploration phase, whereas a low value designates that object '*i*' is in the exploitation phase. This exemplifies the progression from the exploratory to the exploitative stages of the search. The acceleration factor often starts at a high number and diminishes with time. As a result, search agents are aided in their pursuit of the optimal global answer while simultaneously being diverted from less optimal, locally relevant options. In any event, it is worth noting that there may be an unimportant sum of search agents that need a longer exploration period than is typical. In this way, AOA strikes a happy medium between discovery and exploitation.

Step 5: Adjustment of the location: When TF is less than 0.5, location for the next iteration, t + 1, is calculated as (14)

$$x_i^{t+1} = x_i^t + C_1 \times rand \times acc_{i-norm}^{t+1} \times d \times \left(x_{rand} - x_i^t\right) \tag{14}$$

Here C_1 is continuous equals to 2. If $TF > 0.5$, the objects inform their locations using (15).

$$x_i^{t+1} = x_{best}^t + F \times C_2 \times rand \times acc_{i-norm}^{t+1} \times d \times \left(T \times x_{best} - x_i^t\right) \tag{15}$$

$C_2 = 6$, where *C* is a constant. The time variable *T* is defined as T = (C_3) (TF), where C is the constant and TF is the transfer operator, and both variables grow as time passes. *T* grows over time in the interval $\left[C_3, 0.3, 1\right]$ and initially steals a fixed proportion of the top spot. In the beginning, the percentage is low, such that the gap between the best possible position and the present position is wide, making the random walk's steps huge. This proportion increases steadily as the search moves forward to close the distance between the ideal position and the one that is currently occupied. Finding a sound compromise between discovery and abuse is the end result of this.

Here, F is the signal for reversing the direction of motion in (16):

$$F = \begin{cases} +1 & if\ P \le 0.5 \\ -1 & if\ P > 0.5 \end{cases} \tag{16}$$

where $P = 2 \times rand - C_4$.

Step 6: Evaluation: Determine the greatest answer so far by evaluating each item using objective function f. Assign x_{best}, den_{best}, vol_{best}, and acc_{best}.

Table 2 Confusion matrix

	Predicted positive	Predicted negative
Actual positive	True positive (*TP*)	False positive (*FP*)
Actual negative	False negative (*FN*)	True negative (*TN*)

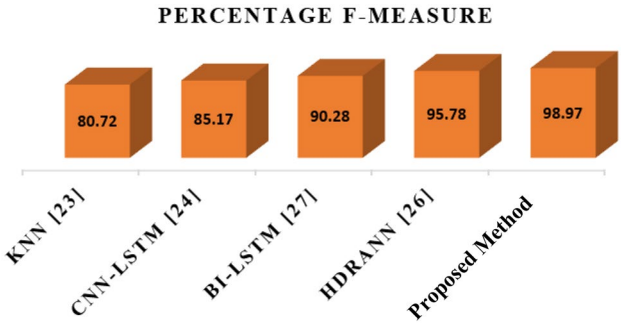

Fig. 4 Percentage F-measure for 60% training data

5 Results and discussion

A Dell G5 gaming PC is used to test the proposed optimized method and compare it to existing deep learning classifiers. A 4.7 GHz Intel Core i7-9700 CPU with turbo technology is installed in the system. The system's memory was a whopping 16 GB of DDR4 RAM. A GB graphics card was added to ensure machine learning procedures. Using the Python programming language, the approach has been included in the Anaconda Navigator.

5.1 Performance measure

The performance indicator is the consistent evaluation of outcomes that provides trustworthy data on the effectiveness of the suggested strategy. In addition, the process of notifying, obtaining, and evaluating information regarding the effects of the assaults is indicative of efficacy. The confusion matrix used for assessing the classifier categorical data and its definition are summarized in Table 2.

True positives are occurrences that have been appropriately labeled as such. The FP implies that the true positive classification is inaccurate. Similar to FN, which are also real negative events that have been misclassified as positive, TN are also true negative events that have been misclassified as false positives.

The mathematical equations for the estimation of the performance parameters are given as:

$$Accuracy(ACC) = (TN + TP)/(TP + TN + FN + FP) \times 100 \tag{17}$$

$$F{-}measure(F{-}M) = 2TP/((2TP + FP + FN)) \times 100 \tag{18}$$

$$Precision\ (PR) = TP/((FP + TP)) \times 100 \tag{19}$$

$$Recall\ (RC) = TP/((FN + TP)) \times 100 \tag{20}$$

The existing techniques, such as KNN [23], CNN-LSTM [24], Bi-LSTM [28], HDRaNN [26], and proposed technique are tested with the dataset, namely DS2OS, to reveal the effectiveness of the proposed model. The results, as obtained with existing and proposed techniques, are shown in Figs. 4, 5, 6, 7, 8, 9, 10 and 11.

Figures 4, 5, 6 and 7 consider the training data of 60%, whereas Figs. 8, 9, 10 and 11 consider the training data of 80% from the given datasets. It is evident from Figs. 4, 5, 6 and 7 that the proposed optimized approach outperforms the other existing techniques. The F-measure of the proposed model is 98.97%. The smallest value of F-measure is provided by KNN [23]. The highest value of F-measure, as obtained with the proposed optimized approach, indicates that the proposed model has a good tradeoff between precision and recall and is hence proficient in detecting and

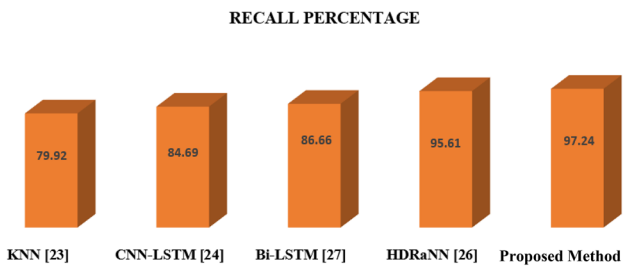

Fig. 5 Percentage recall for 60% training data

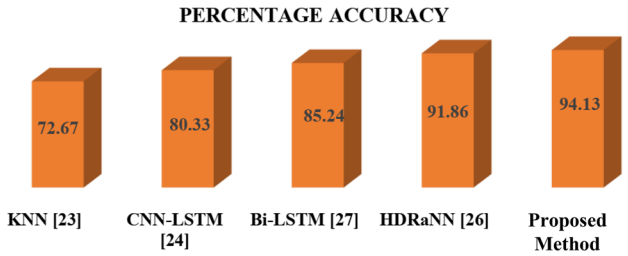

Fig. 6 Percentage accuracy for 60% training data

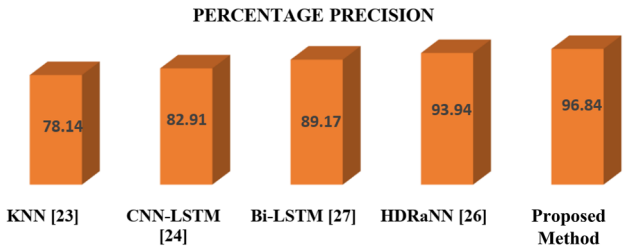

Fig. 7 Percentage precision for 60% training data

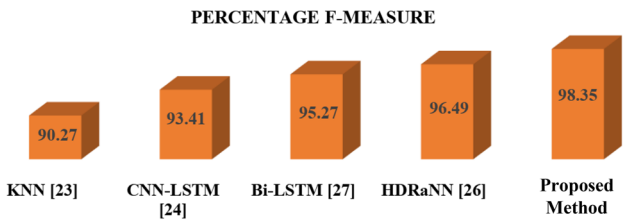

Fig. 8 Percentage F-measure for 80% training data

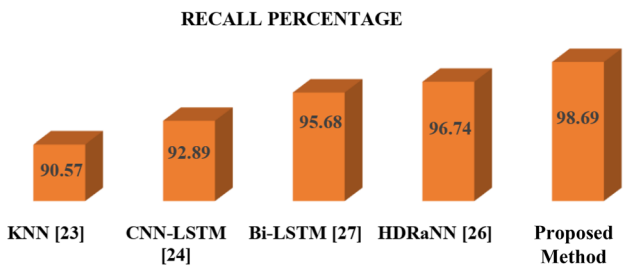

Fig. 9 Percentage recall for 80% training data

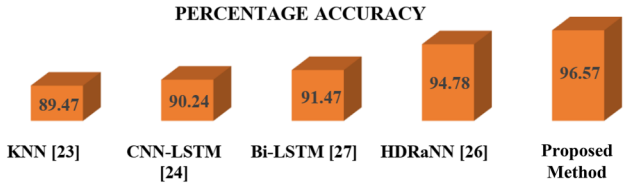

Fig. 10 Percentage accuracy for 80% training data

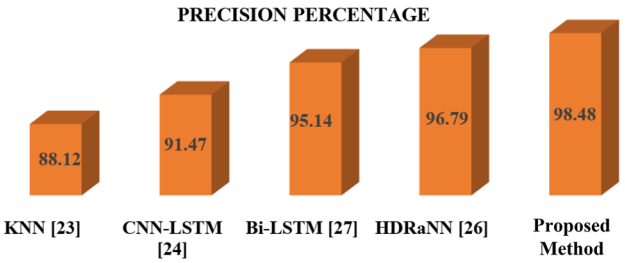

Fig. 11 Percentage precision for 80% training data

classifying assaults with high exactness. Further, the FPR of the proposed system will also be small due to the high value of the F-measure. In a nutshell, it can be said that the high F-measure, as obtained with the proposed optimized approach, will have the following attributes:

1. Improved threat recognition
2. Small FPR
3. Ability to detect genuine assaults
4. Improved reliability
5. Quick detection of the genuine assaults as low value of FPR

Additionally, in order to meet industrial requirements for cyber security, high recall percentages are important. It is also required to avoid legal issues. Further, the high value of the recall percentage is an indicator of the capability of early protection from advanced threats. Hence, the proposed model is also suitable for detecting advanced threats with increased confidence due to the high recall percentage. Further, the F1 scores of the existing techniques and proposed techniques are estimated and shown in Fig. 12.

The index, 'F1-Score', includes precision and recall in the single matrix. This index is often used to reveal the effectiveness of the model. A high value of the F1-score indicates that the model has a high capability of minimizing both errors, namely 'FP' and 'FN'. When the balance between precision and recall is highly desirable, a high F1-score is desirable. It is obvious from Fig. 12 that the proposed model has an F1-score of 0.96, closer to unity, and hence the proposed model provides a more accurate characterization of the assaults than the other existing techniques. Further, from Figs. 8, 9, 10 and 11, it is evident that, by increasing the training and testing values, the performance of the proposed model can be increased. In a nutshell, it can be said that the proposed model is robust, reliable, and has a good tradeoff between recall and precision. Hence, it can be used in industrial applications.

6 Conclusion

A novel pre-training method, using the hybridization of DNN and DAE has been developed in this work for the fast detection of assaults with increased accuracy and a reduced false rate. The proposed method delivers an alternative to deep learning construction replicas through an HPO procedure incorporating the Archimedes optimization algorithm (AOA). The various existing techniques and proposed techniques are tested with the dataset, namely DS2OS, to reveal the effectiveness of the proposed technique. The performance parameters, such as Accuracy, Precision, Recall, F-measure, and F1-score have been estimated with a common dataset. Through the comparative analysis of the results, it is shown that the proposed model provides more accurate results than the other existing techniques. The value of the percentage F-measure of the proposed technique is found to be the highest, i.e., 98.97%. The highest value of F-measure, as obtained

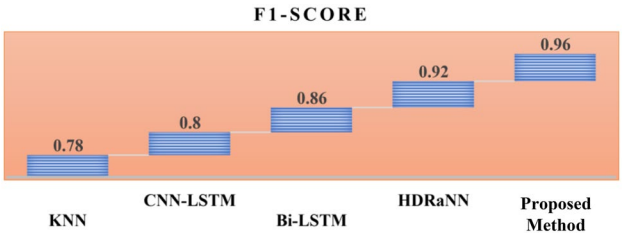

Fig. 12 Outcome's comparison of different techniques for the index, 'F1-Score'

with the proposed model, indicates the proposed model has improved threat recognition with a small FPR and greater reliability. Further, the F1 score of the proposed model is found to be nearly equal to unity. The high value of the F1-score of the proposed model shows the following attributes over the other existing techniques:

1. A good trade-off between recall and precision
2. Higher reliability than the other existing techniques
3. Smaller value of FPR than the other existing techniques
4. Faster detection of assaults than the other existing techniques.
5. More cost-effective than the other existing techniques due to the detection of only real attacks.

Furthermore, increasing the training datasets can enhance the performance of the proposed model. Therefore, taking into account the aforementioned attributes, we can use the proposed optimized approach for industrial applications with good confidence and reliability.

Acknowledgements Not applicable.

Author contributions Dr. Raj Sinha: Supervision, Visualization, Data curation, Formal analysis, Validation. Dr Padmanabh Thakur: Data curation, Supervision, Investigation, Formal analysis, Writing of the original draft. Dr. Sandeep Gupta: Conceptualization, Visualization, Data curation, Investigation, Formal analysis, and validation. Dr. Anand Shukla: Data curation, Formal analysis, Validation.

Funding No funding was received.

Data availability The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: a survey. Appl Sci. 2019;9(20):4396.
2. Khraisat A, Alazab A. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity. 2021;4(18):1–27.
3. Mishra N, Pandya S. Internet of things applications, security challenges, attacks, intrusion detection, and future visions: a systematic review. IEEE Access. 2021;9:59353–77.
4. Wang Z. Deep learning-based intrusion detection with adversaries. IEEE Access. 2018;6:38367–84.
5. Idouglid L, Tkatek S, Elfayq K, Guezzaz A. A novel anomaly detection model for the industrial Internet of Things using machine learning techniques. Radioelectron Comput Syst. 2024;2024(1):143–51.
6. Ibitoye O, Shafiq O, Matrawy A. Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. In: IEEE global communications conference. 2019. p. 1–6.
7. Alsoufi MA, Razak S, Siraj MM, Nafea I, Ghaleb FA, Saeed F, Nasser M. Anomaly-based intrusion detection systems in IoT using deep learning: a systematic literature review. Appl Sci. 2021;11(18):8383.
8. Hassini K, Khalis S, Habibi O, Chemmakha M, Lazaar M. An end-to-end learning approach for enhancing intrusion detection in Industrial-Internet of Things. Knowl Based Syst. 2024;294:111785.
9. Zhang H, Wu CQ, Gao S, Wang Z, Xu Y, Liu Y. An effective deep learning based scheme for network intrusion detection. In: 24th international conference on pattern recognition (ICPR). 2018. p. 682–7.
10. Nimbalkar P, Kshirsagar D. Feature selection for intrusion detection system in Internet-of-Things (IoT). ICT Express. 2021;7(2):177–81.
11. Kasthuri S, Jebaseeli Nisha A. Review on social network analysis in data mining. Infokara Res. 2019;8(12):1168–72.
12. Salloum SA, Alshurideh M, Elnagar A, Shaalan K. Machine learning and deep learning techniques for cybersecurity: a review. In: International conference on artificial intelligence and computer vision. Cham: Springer; 2020. p. 50–7.
13. Booij TM, Chiscop I, Meeuwissen E, Moustafa N, den Hartog FT. ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. IEEE Internet Things J. 2021;9(1):485–96.

14. Asharf J, Moustafa N, Khurshid H, Debie E, Haider W, Wahab A. A review of intrusion detection systems using machine and deep learning in Internet of Things: challenges, solutions and future directions. Electronics. 2020;9(7):1177.
15. Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. J Inf Secur Appl. 2020;50:102419.
16. Zhou X, Liang W, Li W, Yan K, Shimizu S, Kevin I, Wang K. Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet Things J. 2021;9(12):9310–9.
17. Latif S, Zou Z, Idrees Z, Ahmad J. A novel attack detection scheme for the Industrial Internet of Things using a lightweight random neural network. IEEE Access. 2020;8:89337–50.
18. Chegu S, Reddy GU, Bhambore BS, Adeab K, Honnavalli P, Eswaran S. An improved filter against injection attacks using regex and machine learning. Netw Secur. 2022;2022(9):1353–4858.
19. Li B, Wu Y, Song J, Lu R, Tao Li, Zhao L. DeepFed: federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Trans Ind Inf. 2020;17(8):5615–24.
20. Maheswari VU, Aluvalu R, Kantipudi MP, Chennam KK, Kotecha K, Saini JR. Driver drowsiness prediction based on multiple aspects using image processing techniques. IEEE Access. 2022;10:54980–90.
21. Srastika N, Bhandary SR, Honnavalli SP, Eswaran S. An enhanced malware detection approach using machine learning and feature selection. In: 2022 3rd international conference on electronics and sustainable communication systems. 2022. p. 909–14.
22. Mendonca RV, Silva JC, Rosa RL, Saadi M, Rodriguez DZ, Farouk A. A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms. Expert Syst. 2022;39(5):12917.
23. Guezzaz A, Benkirane S, Mohyeddine M, Attou H, Douiba M. A lightweight hybrid intrusion detection framework using machine learning for edge-based IIoT security. Int Arab J Inf Technol. 2022;19(5):822–30.
24. Rani S, Singh A, Elkamchouchi DH, Noya ID. Lightweight hybrid deep learning architecture and model for security in IIOT. Appl Sci. 2022;12(13):6442.
25. Awotunde JB, Chakraborty C, Adeniyi AE. Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. Wirel Commun Mob Comput. 2021;2021:1–17.
26. Huma ZE, Latif S, Ahmad J, Idrees Z, Ibrar A, Zou Z, Alqahtani F, Baothman F. A hybrid deep random neural network for cyberattack detection in the Industrial Internet of Things. IEEE Access. 2021;9:55595–605.
27. Soliman S, Oudah W, Aljuhani A. Deep learning-based intrusion detection approach for securing Industrial Internet of Things. Alex Eng J. 2023;81:371–83.
28. Almaiah MA, Ali A, Hajjej F, Pasha MF, Alohali MA. A lightweight hybrid deep learning privacy preserving model for FC-based industrial internet of medical things. Sensors. 2022;22(6):2112.
29. Malhotra P, Singh Y, Anand P, Bangotra DK, Singh PK, Hong WC. Internet of Things: evolution, concerns and security challenges. Sensors. 2021;21(5):1809.
30. AL-Hawawreh M, Moustafa N, Sitnikova E. Identification of malicious activities in Industrial Internet of Things based on deep learning models. J Inf Secur Appl. 2018;41:1–11.
31. Ge M, Syed NF, Fu X, Baig Z, Robles-Kelly A. Towards a deep learning-driven intrusion detection approach for Internet of Things. Comput Netw. 2021;186:107784.
32. Pahl MO, Aubet FX. Ds2Os traffic traces IoT traffic traces gathered in a the Ds2Os IoT environment. 2018. [Online]. https://www.kaggle.com/francoisxa/ds2ostraffictraces.
33. Ma T, Wang F, Cheng J, Yu Y, Chen X. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. Sensors. 2016;16(10):1701.
34. Gupta S, Varshney T. The future of IoT with automation in engineering and modern technology. Series: computer science & media. Hauppauge: Nova Publisher; 2024.
35. Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell. 2021;51:1531–51.