frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 11 April 2023
DOI 10.3389/fcomp.2023.997159

OPEN ACCESS

EDITED BY
Antonino Rullo,
University of Calabria, Italy

REVIEWED BY
Raj Chaganti,
Toyota Research Institute (TRI), United States
Zhifeng Xiao,
Penn State Erie, The Behrend College,
United States
Zhihong Tian,
Guangzhou University, China

*CORRESPONDENCE
Yakub Kayode Saheed
✉ yakubu.saheed@aun.edu.ng

SPECIALTY SECTION
This article was submitted to
Computer Security,
a section of the journal
Frontiers in Computer Science

RECEIVED 18 July 2022
ACCEPTED 24 January 2023
PUBLISHED 11 April 2023

CITATION
Saheed YK, Usman AA, Sukat FD and
Abdulrahman M (2023) A novel hybrid
autoencoder and modified particle swarm
optimization feature selection for intrusion
detection in the internet of things network.
Front. Comput. Sci. 5:997159.
doi: 10.3389/fcomp.2023.997159

A novel hybrid autoencoder and modified particle swarm optimization feature selection for intrusion detection in the internet of things network

Yakub Kayode Saheed[1,2]*, Aisha Abubakar Usman[1,2], Favour Dirwokmwa Sukat[1,2] and Muftahu Abdulrahman[3,4]

[1]School of Information Technology and Computing, Yola, Adamawa State, Nigeria, [2]American University of Nigeria, Yola, Adamawa State, Nigeria, [3]African Centre of Excellence on Technology Enhanced Learning, Abuja, Nigeria, [4]National Open University of Nigeria, Abuja, Nigeria

The Internet of Things (IoT) represents a paradigm shift in which the Internet is connected to real objects in a range of areas, including home automation, industrial processes, human health, and environmental monitoring. The global market for IoT devices is booming, and it is estimated that there will be 50 billion connected devices by the end of 2025. This explosion of IoT devices, which can be expanded more easily than desktop PCs, has led to an increase in cyber-attacks involving IoT devices. To address this issue, it is necessary to create novel approaches for identifying attacks launched by hacked IoT devices. Due to the possibility that these attacks would succeed, Intrusion Detection Systems (IDS) are required. IDS' feature selection stage is widely regarded as the most essential stage. This stage is extremely time-consuming and labor-intensive. However, numerous machine learning (ML) algorithms have been proposed to enhance this stage to boost an IDS's performance. These approaches, however, did not produce desirable results in terms of accuracy and detection rate (DR). In this paper, we propose a novel hybrid Autoencoder and Modified Particle Swarm Optimization (HAEMPSO) for feature selection and deep neural network (DNN) for classification. The PSO with modification of inertia weight was utilized to optimize the parameters of DNN. The experimental analysis was performed on two realistic UNSW-NB15 and BoT-IoT datasets that are suitable for IoT environment. The findings obtained by analyzing the proposed HAEMPSO against the Generic attack in the UNSW-NB15 dataset gave an accuracy of 98.8%, and a DR of 99.9%. While the benign class revealed an accuracy of 99.9% and DR of 99.7%. In the BoT-IoT dataset, the DDoS HTTP attack revealed an accuracy of 99.22% and DR of 97.79%. While the benign class gave an accuracy of 97.54% and DR of 97.92%. In comparison with the state-of-the-art machine learning schemes, our proposed HAEMPSO-DNN achieved a competitive feat in terms of DR and accuracy.

KEYWORDS

autoencoder (AE), Internet of Things, machine learning, particle swarm optimization, deep neural network

1. Introduction

The Internet of Things (IoT) is a unique concept in the field of computer networking that enables the communication of various types of objects *via* the Internet (Fenanir et al., 2019). These devices can be RFID tags, actuators, sensors, or mobile phones; all of which connect and cooperate utilizing a single addressing technique (Abbas et al., 2019). The IoT allows the seamless

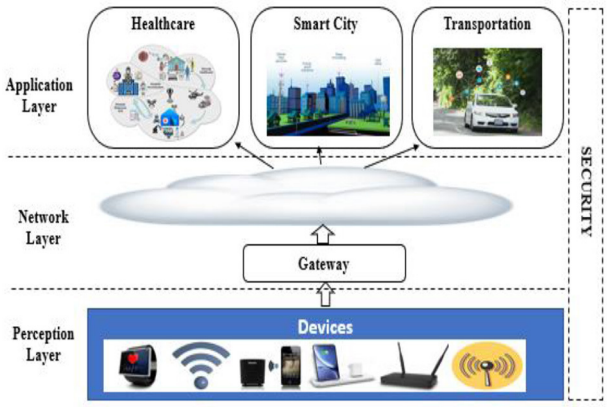

FIGURE 1
The Internet of Things design pattern (Fenanir et al., 2019).

integration of all kinds of infrastructures, at all times, for all of us, and on any object, as a result of ubiquitous computing (Atzori et al., 2010). The IoT is a network of numerous networked things that enables people and objects to communicate and build intelligent environments in areas such as transportation, agriculture, healthcare, energy, and cities (Marlow, 2018). The IoT architecture model depicted in Figure 1 is comprised of three layers; the application, the network, and the perception layers (Leo et al., 2014). The IoT is a network of sensors-equipped objects or things. Sensors are placed on goods or things and the Internet (usually wirelessly) to collect and exchange data in the IoT (Askarzadeh, 2016).

IoT technologies present various previously unimagined prospects for human interconnection. Hypertext transfer protocol, message queuing telemetry transport (MQTT), and domain name system are the internet protocol that runs IoT services under the TCP/IP architecture (Saheed, 2022a). IoT has the potential to improve people's lives and also contribute to the development of the smart city. The IoT provides a centralized stage for sharing of information across the industry, environment, and society (Singh et al., 2020). As a result, efficient coordination between promised services and accessible resources is required for IoT-enabled intelligent cities to function effectively (Arshad et al., 2020). Due to the exponential growth of IoT devices, we require improved cellular network architecture to deliver high-speed connections among such IoT gadgets (Thamilarasu and Chawla, 2019). Mid-band, low-band, and millimeter-wave frequency ranges enhance the rate at which data is transferred over multi-hop connections in these designs. The fast growth in the quantity and variety of intelligent devices linked to the Internet has involved intruders looking to disseminate hateful content over this network of things (Subham et al., 2022). Intrusion behavior refers to an effort to violate the integrity, confidentiality, and access to specified resources, and it is the core focus of the Internet of Things Security Research Institute (Sicari et al., 2020). By gaining access to users' personal information, spying, and causing financial losses, attackers may jeopardize their privacy and security. Apart from compromising the security of the IoT, these intrusions pose a threat to the entire ecosystem, including applications, sites, servers, and social media *via* controlled smart objects such as robot networks (Li et al., 2020).

The IoT gadgets are regularly organized in an unsafe and hostile environment, increasing their susceptibility to various threats (Sedjelmaci et al., 2016). As a result, security solutions are critical for defending IoT gadgets against intruder attacks (Kayode Saheed et al., 2022). An Intrusion Detection System (IDS) is a tool that analyzes the activities and events of a system or network to detect attacks against them (Raza et al., 2013). It may serve as a secondary line of protection against intruders (Raza et al., 2013). The basic goal of an IDS is to accurately identify as many attacks as possible while utilizing the minimal energy feasible in resource-constrained circumstances (Liang et al., 2020).

Surprisingly, IDS is recommended as an excellent tool for monitoring network action, assisting in determining unauthorized use, detecting system damage, and protecting systems from internal and external invasions (Kayode Saheed et al., 2022). IDS can be categorized into two types based on the monitored environment: host-IDS (HIDS) and network-IDS (NIDS)(Habib et al., 2020). The latter are strategically located throughout the network, to cover all susceptible points. They must examine network circulation by inspecting the payload and header fields of each packet. Their composition is determined by the sensitivity of the application. There is always a chance that an event will be misclassified as negative due to configuration issues or the existence of encrypted traffic. HIDS continuously collects and evaluates all organizational network gadgets to identify insider threats (Saheed et al., 2022a). They take a photo of existing system logs and compare them to older ones to spot anything unexpected (Ahanger et al., 2022). If something, suspicious occurs, HIDS initiates the appropriate action based on its programming (Subham et al., 2022). Thus, HIDS gain a better understanding of internal traffic flow and are utilized to protect against any intrusion that might be hidden by a NIDS during the initial phase.

Network security is the primary concern in IoT networks because the majority of manufacturers do not prioritize security standards throughout the design. With its rapid expansion in numerous domains like wearable gadgets, smart sensors, and home appliances, the IoT is poised to touch numerous facets of our lives. The pervasiveness, connectivity, and limited processing capability of IoT devices define them.

As a result of the limits of IoT technologies, security has emerged as a critical concern for IoT networks and services. The Internet of Things devices are small, varied, and lack compatibility (Atlam et al., 2022). These qualities broaden the attack surface and complicate the development of any security solution (Ramadan and Yadav, 2020). Not only are IoT devices susceptible to network attacks (Putra et al., 2020), but also powerful hackers from unauthorized internet users. Cryptography algorithms are presented in some literature for IoT authentication and confidentiality. However, cryptography techniques are computationally and time-intensive, making them unsuitable for IoT devices (Minh Dang et al., 2019). Additionally, cryptography techniques aid in the authenticity and integrity of networks and data. Additional technologies are critical to measuring IoT network traffic to avoid the current network attacks. To maintain such functionality, an IDS is critical. IDS are responsible for monitoring, analyzing, and detecting network attacks.

The literature discusses a variety of IDS strategies. These strategies are classified as either anomaly or signature (Blanco et al., 2019). The signature-based detection method is founded on an account of pre-defined harmful activity patterns (Saheed et al., 2022b), whereas the anomaly detection approach is based on the observation of deviations from normal behavior to indicate intrusions (Saheed and Hamza-usman, 2020). As a result, the strategy based on anomalies was capable of detecting unknown attacks that did not follow predetermined activity patterns (Saheed, 2022b). We provide an anomaly-based IDS methodology for IoT networks in this paper.

Numerous solutions have been developed in the past that utilize Machine Learning/Deep Learning methodologies to detect/prevent invasions. Numerous of them placed a lesser emphasis on the pre-processing stage, which is largely concerned with feature selection. Therefore, it is possible to witness a direct effect on the classifier of the supplied algorithm. Additionally, the time needed for training the model is increased as a result of the insufficient pre-processing stage. The training time is also increased as a result of the current neural network algorithm's back-propagation technique. This work presents a hybrid optimization and Deep Neural Network-based (DNN)-based better IDS model for an IoT-enabled intelligent approach. This paper addresses the aforementioned difficulties by presenting two techniques for feature selection and a DNN for the model classification. In this research, we propose a novel hybrid Autoencoder and Modified Particle Swarm Optimization (HAEMPSO) for feature selection and deep neural network (DNN) for classification. The PSO with modification of inertia weight was utilized to optimize the parameters of DNN. The experimental analysis was performed on two realistic UNSW-NB15 and BoT-IoT datasets that are suitable for IoT environment.

The following are the paper's main contributions:

- A hybrid Autoencoder with the modification of Particle Swarm Optimization (HAEMPSO) for feature selection is proposed in this research.
- To minimize information leakage on the test set, an effective hybrid IDS is constructed using the minimum-maximum normalization strategy for feature scaling.
- The HAEMPSO approach combines autoencoder and modified particle swarm optimization (global best position and self-best position) to prioritize the features that contribute the most to the output, hence resulting in a lightweight model.
- Additionally, a model based on Deep Neural Networks is developed to improve classification accuracy.
- Leverages the realistic UNSW-NB15 and BoT-IoT datasets that reflect modern-day attacks in the IoT ecosystem.
- The proposed hybrid feature selection DNN model outperformed the traditional ML algorithms.

The remainder of the paper is structured as follows. The relevant work was discussed in Section Related work. The approach is then presented in Section Materials and methodology, followed by the evaluation results obtained through experiments and comparative studies in Section Proposed HAEMPSO-DNN model. Finally, Section Results and discussion has the conclusion.

2. Related work

Thanigaivelan et al. (2016) provided a brief overview of networked anomaly detection for the IoT. The suggested IDS works on the premise of detecting network anomalies by analyzing the properties of one-hop nearest neighbors such as packet size and data rate.

Pongle and Chavan (2015) demonstrated the capability of an intrusion detection system (IDS) to identify wormhole attacks in IoT gadgets. The researcher shoulder that successful wormhole assaults always leave traces on the system, such as a huge volume of control packets transmitted between the two ends of the tube or the establishment of a large number of neighbors following the attack. Basis on this reason, the researchers presented three ways of detecting such network anomalies. According to their research, the approach yields a true positive of 94% for wormhole attacks and 87% for identifying both the compromise and the attacker. However, no information is provided regarding the rate of false positives. Additionally, the scientists examined the power and memory consumption of the nodes. Due to its low power and memory consumption, the suggested system appears to be well-suited for IoT devices. On the other hand, the data collected should be compared to published literature to establish a baseline.

Aydin et al. (2009) showed a hybrid intrusion detection system (HIDS) by integrating two approaches with the Snort signature-based IDS. The proposed system is assessed using the IDEVAL data set, which reveals a significant upsurge in the number of assaults spotted when compared to signature-based systems when employing the proposed hybrid IDS.

Wang et al. (2010) presented the FC-ANN approach for intrusion detection, which is founded on the fuzzy clustering method and ANN. The FC-ANN technique is composed of three major modules: ANN, fuzzy aggregation, and fuzzy clustering. The fuzzy clustering function is utilized to create clusters from a set of given data. The ANN component is utilized to discover the design associated with each subgroup. The FC-ANN technique was evaluated on the KDD CUP'99 data and showed efficacy against low-frequency attacks.

The study (Govindarajan and Chandrasekaran, 2011) presented a hybrid neural-based IDS design that utilizes two techniques: MLP and RBF. To improve robustness, accuracy, and overall generalization, hybrid modeling approaches are applied, including bagging classifiers. Additionally, this study makes use of UNM Send-Mail Data, which is founded on a University of New Mexico-developed system. The proposed IDS attained an accuracy of 98.88% for normal traffic and 94.31% for abnormal traffic, outperforming the classifiers that comprise it.

Chung and Wahid (2012) address the issue of decision rule generation by combining an IDS-RS feature selection strategy with an SSO–WLS data categorization technique. By weighing three specified constants, the study proposes a comprehensive system approach for optimizing the search process in SSO rule mining. The new results on the KDD CUP 1999 data indicate that the presented method for NIDS utilizing an IDS-RS set achieves a 93.3% accuracy rate in an increase of 20 runs.

Elbasiony et al. (2013) established a hybrid architecture for combining misuse and anomaly detection using two methods:

(1) the *K*-means algorithm and (2) RF. Precisely, this system detects abuse intrusions using the RF approach and anomalies utilizing the *k*-means algorithm. Because RF makes use of correlated variables, this framework exhibits low-performance degradation and interpretability.

Kim et al. (2014) connect a model for detecting misuse with one for detecting anomalies in a breakdown structure. This inquiry makes use of the C4.5 model technique and many one-class SVMs. The NSL-KDD data experiments reveal that the presented HIDS strategy outperforms typical intrusion detection techniques in terms of testing time, training time, and detection performance. Li et al. (2021a) suggested a machine learning-based classification approach for attribution companies utilizing APT malware in IoT. To more effectively identify advanced persistent threat attack activities and safeguard the security of IoT, the authors try to identify the actual attacking organization entities. The approach could locate the group responsible for sophisticated APT assaults against IoT products and services. The SMOTE-RF model works well and has steady performance when classifying APT malware, and the approach of feature extraction provided obtained more than 80% accuracy in general models.

An ensemble model was presented by Li et al. (2021b) for identifying fraudulent mining codes on cloud platforms. For the classification, they used bagging and boosting algorithms. The experimental findings demonstrate that the values of AUC and F1-score for the provided dataset can approach 99.2 and 98.7%, respectively.

Liu et al. (2021) provide a gradient descent-based particle swarm optimization method (PSO-LightGBM) for IoT intrusion detection. This technique employs PSO-LightGBM to extract the data's features before feeding it into one-class SVM (OCSVM) to find and recognize harmful data. The intrusion detection model is tested using the UNSW-NB15 dataset. The experimental findings demonstrate that the suggested model is extremely reliable in identifying both legitimate and different types of harmful data with an accuracy of 86.68%.

An IDS for IoT was suggested by Alterazi (2022). This article investigates various performance-based AI models to accurately forecast attacks and issues with IoT devices. The efficiency of the recommended method was demonstrated using ant colony optimization, genetic algorithms, and particle swarm optimization (PSO). The findings of the suggested method using PSO performed about 73% better than those of the current systems.

The study (Lin et al., 2015) introduced CANN, a technique for merging nearest neighbors and cluster centers. On the KDD CUP'99 dataset, experimental results indicate that the CANN technique outperforms SVM and *K*-NN classifiers. Thanigaivelan et al. (2016) suggested a classifier technique based on a mixture of three algorithms, namely the C4.5 DT algorithm, the NBTree algorithm, and the random tree algorithm.

Pongle and Chavan (2015) introduced a HIDS in by combining extreme learning machines, *K*-means clustering, and support vector machines. Experiments on the KDD CUP'99 data indicate that the hybrid NIDS has the potential to significantly increase performance and attain an accuracy of 95.75 percent.

Aydin et al. (2009) developed the HCPTC-IDS IDS system, which is founded on the aggregate probability of predictions from a large number of algorithms. The HCPTC-IDS system consists of two layers: (1) a tree of learners; and (2) a final classification that integrates the first layer's multiple probability predictions. Experiments with NSL-KDD and KDD'99 demonstrate that the HCPTC-IDS system outperforms previous similarity-based intrusion detection systems, with an accuracy of 96.27% for KDD'99 and 89.75% on NSL-KDD.

The study (Wang et al., 2010) proposed a method for finding anomalies that incorporate both an SVM and a GA to enhance the classification performance of SVMs. Utilizing the KDD CUP 1999 data, the experimental findings suggest an extraordinary true-positive of 97.3% and a false-positive of 0.017.

Derhab (2019) introduced the RSL-KNN IDS for detecting simulated command attacks. The system is based on a technique called random subspace learning and the KNN algorithm. The RSL-KNN scheme is used in conjunction with blockchain to identify any manipulation of the OpenFlow regulation in real time.

An autoencoder model for intrusion detection in an IoT environment was proposed by Lahasan and Samma (2022). The model has a shallow architecture, few hidden neurons, and few input features. On the NBaIoT dataset, classification was performed using the KNN technique. The proposed model obtained anomaly detection accuracy of 99%, according to experimental results. The limitation of this study is that the autoencoder used is computationally expensive.

An autoencoder strategy for intrusion detection in the Industrial Internet of Things was suggested by Zhang and Zhang (2022). Dimensionality reduction was performed using the sparse autoencoder. With an accuracy of 95.42 percent, the experiment has shown that the suggested method has good network attack identification and detection capability.

Ferrag and Maglaras (2020) established a unique energy framework based on blockchain and DL, dubbed DeepCoin, for smart grids. The architecture of DeepCoin employs two distinct schemes: one based on deep learning and another on the blockchain. The deep learning-based technique detects network threats *via* RNN, whereas the blockchain-based method detects fake transactions. We recommend the reader to the recent work in Ferrag et al. (2019a) for additional details on DL methods used in cyber security intrusion detection. Table 1 summarizes exemplary studies on hybrid IDS, which incorporate ML, DL, and DM techniques. Additionally, it discusses the security problem that each of these solutions attempts to address, as well as the dataset used to assess their performance.

The majority of relevant papers make use of the old KDD'99 and NSL-KDD sets of data, which are of incomplete real-world relevance for a current IDS. Since these datasets were produced in 1999, both malicious and innocuous network traffic has evolved significantly, and the results generated from them are typical of limited utility. To address various limitations of previously published approaches, such as low detection rate of infrequent attacks, misclassification of attacks, and accuracy. We offer a novel hybrid intrusion detection system (IDS) that incorporates two distinct models, namely Autoencoder and Particle Swarm Optimization. Additionally, we utilized the BoT-IoT (Shafiq et al., 2020) and UNSW-NB15 (Choudhary and Kesswani, 2020) datasets, which we divide into training and testing sets, to assess their effectiveness in noticing network intrusions and to compare it to the performance of other machine learning approaches given by earlier researchers.

TABLE 1 Summary of existing studies based on hybrid IDS.

References	Cyber technique	DM and ML methods	Dataset utilized	Results
Aydin et al. (2009)	Hybrid IDS, which combines anomaly-based and signature-based IDS	Detection of packet header anomalies-Detection of abnormal network traffic	IDEVAL	High DR
Wang et al. (2010)	Hybrid IDS, in which the results are aggregated using the fuzzy aggregation module	ANN fuzzy clustering	KDDCup 99	FC-ANN: Accuray = 96.71%; BPNN: Accuracy = 96.65%
Govindarajan and Chandrasekaran (2011)	Neural-based hybrid IDS	MLP neural network RBF neural network	UNM-Send mail data	MLP: Accuracy = 98.88%; RBF: Accuracy = 94.21%
Chung and Wahid (2012)	Hybrid IDS	Dynamic swarm intelligence Reduced swarm optimization	KDD Cup 99	SSO–WLS: Accuracy = 93.3%; SSO: Accuracy = 89.6; PSO: Accuracy = 88.5%
Elbasiony et al. (2013)	Creating a hybrid architecture by combining abuse and anomaly intrusion detection	Algorithm of random forests Algorithm of *K*-means clustering	KDD Cup 99	Abuse: DR = 92.73%; Abuse: FPR = 0.54; Anomaly: DR = 95%; Anomaly:FPR = 6.3
Kim et al. (2014)	Creating a hybrid architecture by combining abuse and anomaly detection	Algorithm for C4.5 decision trees SVM model	NSLKDD	C4.5: Training time = 21.37; SVM: training time = 403.27
Liu et al. (2021)	Particle swarm optimization for feature selection in IoT	One class SVM for classification	UNSW-NB15	Accuracy = 86.68%
Alterazi (2022)	PSO for feature dimensionality selection in IoT	Genetic algorithm and Ant Colony Optimization	NSLKDD	Accuracy = 73%
Lin et al. (2015)	Consolidating cluster nodes and their nearest neighbors	CANN and KNN classifiers	KDD Cup 99	CANN: Accuracy = 99.76%; DR = 99.99%, FPR = 0.00; KNN: Accuracy = 80.6%, DR = 80.32%, FAR = 99.92
Aslahi-Shahri et al. (2016)	Hybrid IDS	GA for feature selection and SVM for classification	NSLKDD	SVM = 9.31%; Precision = 94.1%; Recall = 93.1%
Kevric et al. (2017)	Combining tree-based classifier models	C4.5 Random Tree NB	KDD Cup99	C4.5: Accuracy = 79.15%; Random Tree = 71.46%; NB = 75.54%
Ferrag et al. (2019b)	Hybrid IDS	SVM ELM K-means clustering	KDD Cup 99	High accuracy and precision
Ferrag and Maglaras (2020)	Hybrid IDS, combining RNN with blockchain technology	RNN	BoT-IoT, CICIDS2017, and Power system	DDoS Accuracy = 99.89%; Heartbleed = 100%; XSS = 91.75%
Derhab (2019)	The HIDS which adds random subspace learning with blockchain	Random Learning Subspace	Power system	RSL-KNN: Accuracy = 90.08%, FPR = 0.3
Ferrag and Maglaras (2019)	A HIDS which combines DL approaches with blockchain technology	Deep learning	CIC-IDS2018	Improved accuracy
Ferrag et al. (2020)	Adding tree-based classifier methods to create a hybrid IDS	Random forest JRip classifier REP Tree	CICIDS2017	RDTIDS Accuracy = 96.66%, DR = 96.995%
Lahasan and Samma (2022)	Autoencoder for feature selection in IoT	KNN	N-BaIoT	Accuracy = 99%
Zhang and Zhang (2022)	Autoencoder for feature dimensionality reduction in IIoT	Softmax	NSLKDD	Accuracy = 95.42
Ravi et al. (2022)	Meta classifiers in IoT	KPCA, RNN	UNSW-NB15	Accuracy = 99%, Precision = 99%, Recall = 99%, and F1-score = 99%
Chohra et al. (2022)	Chamelleon	RF, XgBoost	NSLKDD UNSW-NB15	UNSW-NB: Accuracy = 89.52%, NSLKDD:Accuracy = 90.1%

2.1. The motivation for the present study

Security remains the primary concern while developing IoT applications, as it cannot be overlooked due to the interconnectedness and sensitivity of the acquired data. Additionally, the IoT is restrained by several limits in terms of devices and components, such as the limited capacity for processing, power consumption, and memory, as well as by the IoT's heterogeneous and ubiquitous nature, compounding the concerns (Oh et al., 2014). As a result, implementing a security policy for these systems is critical. Along with the establishment of firewalls and increasingly sophisticated authentication methods, such security policies must be complemented with monitoring technologies that audit the information system and detect potential intrusions. IoT networks become possible due to advancements in sensing technologies. However, IoT devices have a variety of restrictions, including restricted energy sources and capabilities. Additionally, typical cryptography and IDS approaches may be incompatible with such a network. Moreover, as more people gain access to the Internet, hacking tactics become more sophisticated and accessible (Ramadan and Yadav, 2020). As a result, establishing an efficient monitoring strategy for intrusion detection presents a difficulty. This results in a variety of research suggestions aimed at improving the performance of IoT intrusion detection. NSL-KDD cup is a well-known dataset that has been widely explored. It evolved into a de facto standard for evaluating new algorithms. Regrettably, the existing approaches have the following shortcomings:

- Minimum attack categorization rate
- The minimum attack detection rate
- Overhead time
- The feature selection stage is costly as a result of the huge dimensional data and
- Minimal accuracy.

Thus, the issue at hand is to propose an efficient IDS approach that solves the following issues where the detection rate is critical, particularly for the IoT runtime process. Due to the massive dimension of the data, the feature selection process is time-consuming. Additionally, accuracy is a concern when it comes to IoT devices being employed in mission-critical areas such as military systems or healthcare. In this research, we used the minimum-maximum technique for feature scaling and a unique HAEPSO for feature selection. There are, however, additional feature scaling and selection strategies, such as the *z*-score methodology and LDA. While the *z*-score technique does not necessarily require knowledge of the standard deviation, the LDA technique is extremely sensitive to outliers. As a result, we decided to preprocess the data using minimum-maximum and to choose features in the second step using a new hybrid AE-PSO. The final phase entails classifying the data with the help of the DNN.

3. Materials and methodology

We present the proposed system, autoencoder, particle swarm optimization, HAEPSO, and DNN in this section.

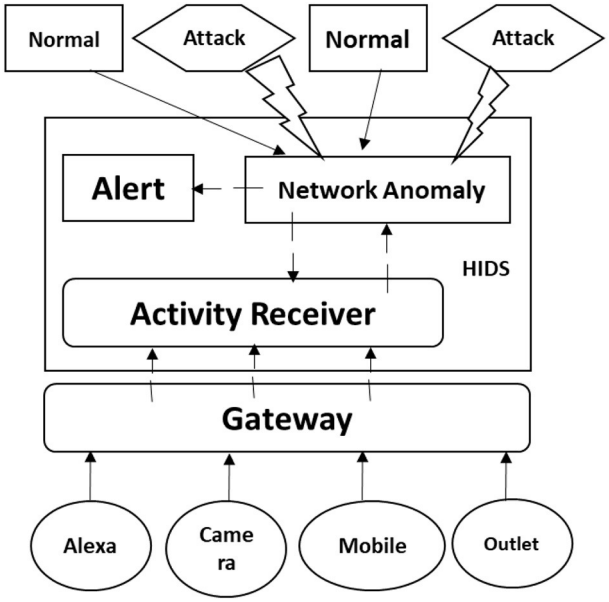

FIGURE 2
The framework of our proposed HIDS model.

3.1. Proposed system

One of our key objectives is for the IDS to be compliant and lightweight with the processing capability of the constrained nodes. As a result of the limited processing capabilities and power consumption of an IoT node, an active intrusion detection sensor cannot be deployed in each node. As a result, we built a central IDS design to address the issues of diminished capabilities on one hand and peripheral diversity on the other, with the IDS running on the IoT's network above the Gateway element. The activity diagram in Figure 2 depicts our Hybrid Intrusion Detection System (HIDS), which identifies intrusions by comparing observed behavior to expected behavior. An alarm is raised if the two behaviors diverge. It is divided into three stages:

a. Activity Receiver: During this phase, the HIDS component activity receiver collects and records all IoT devices (such as Alexa with IP address = 111.30.42.59.70; camera with IP address = 128.145.78; Miscellaneous with IP address = 216.46.84.91; mobile with IP address = 45; and outlet with IP address = 222.67) activities to construct the current actions, which will be presented as a feature space.
b. Network Anomaly Detection: The detection stage is responsible for analyzing and detecting intrusions. It is the central aspect of our HIDS.
c. Alert: Following the identification of an attack, the suggested system stops the user and terminates his session, before alerting the administration to take appropriate action.

3.2. Autoencoder

The AE is a three (3) layer unsupervised neural network (NN) with an input layer, a hidden layer, and an output layer. Figure 3

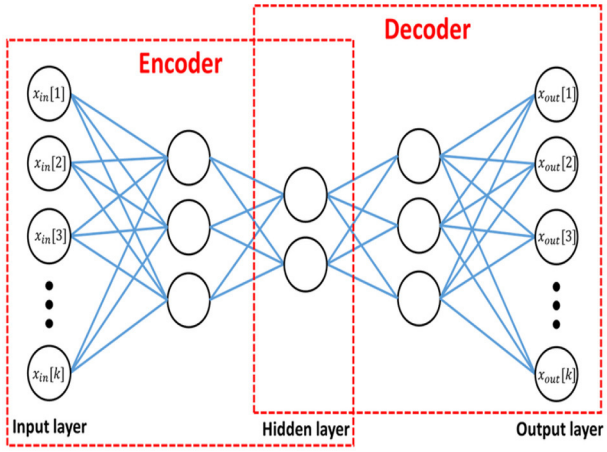

FIGURE 3
The basic structure of an autoencoder.

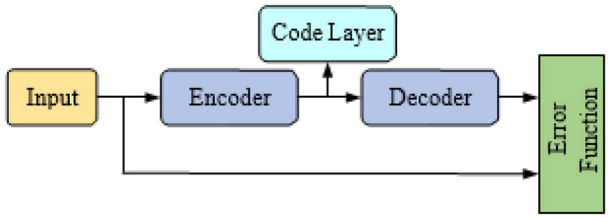

FIGURE 4
Representation of an autoencoder (Yan and Han, 2018).

illustrates the general organization of AE, whereas the representation is depicted in Figure 4.

The AE may gradually change individual feature vectors into generic feature vectors, thus realizing the nonlinear transition from a high to low-dimension data space. The automatic encoder's operation can be divided into two different phases (Yan and Han, 2018): encoding and decoding, which can be characterized as follows: The method of encoding from the input *via* the hidden layer is as follows:

$$E = f\theta_1(K) = \sigma(W_{xy}K + \varnothing_1) \quad (1)$$

The method of decoding from the hidden to the reconstruction layer is as follows:

$$M = f\theta_2(K) = \sigma(WxyK + \varnothing_2) \quad (2)$$

In the formulas above, $K = (k, k, \ldots, k_n)$ is the data input vector, $M = (m_1, m_2, \ldots m_n)$ is the vector reconstruction of the data input, and $E = (e, e, \ldots, e)$ is the dimensional low vector output from the hidden layer, $K \in S^n$, $M \in S^n$, $E \in S^n$ (n is the input dimension vector and m is the units hidden).

$W_{xy} \in S^{nxm}$ is the weight matrix for the linking between the layer input and the layer hidden. $W_{yu} \in S^{nxm}$ is the connection weight matrix between the output and layer hidden. To rebuild the data input with the greatest accuracy feasible while minimizing resource usage during model training, $W_{yu} = W_m^T$ typically is in the experiments. $\varnothing_1 \in S^{nxm}$ and $\varnothing_2 \in S^{nxm}$ are the vector basis of the hidden and input layer. $f\theta_1(.)$ and $f\theta_2$ (.) are the functions used for activation of the output neurons and input neurons, whose roles are responsible for mapping the outcome of the network summing to [1, 0]. In this investigation, we used the sigmoid activation function as the activation function.

$$f\theta_1(.) = f\theta_2(.) = \frac{1}{1 + e - x} \quad (3)$$

The discrepancy between both the output rebuilt data and the original data can be reduced by adjusting the encoder and decoder settings, indicating that AE reassembles the actual data *via* training. At this point, we believe that the data produced by the hidden layer units is the best low-dimensional approximation of the source data, as it contains all of the information included in the original data. Between H and Y, the reconstruction error function $Y_e(W, \theta)$ uses the mean squared error (MSE) function as specified in formula 4, where N is the number of samples input. The autoencoder was used for feature selection to select the significant feature from the two datasets, then the selected features are then passed into DNN. As a result of the slow performance of DNN, we then use the PSO with the modification of inertia weight to optimize the parameters of DNN.

$$Y_e(W, \theta) = \frac{1}{2N} \sum_{s-1}^{N} |Y - X|^2 \quad (4)$$

3.3. Particle swarm optimization

PSO is an evolutionary and optimization technique inspired by nature that is used to address computationally difficult optimization problems (Derhab, 2019). PSO is a robust technique inspired by swarm behavior and intelligence (Lahasan and Samma, 2022). It was created by James Kennedy and Russ Eberhart in 1995. PSO is a computer method for optimizing problems by iteratively attempting to improve candidate solutions in terms of a specified quality metric (Zhang and Zhang, 2022). A particle is a candidate solution, and the search space is improved by shifting the particles around. The velocity and position of each particle are prejudiced by its best-known position, which is updated in each iteration by better positions discovered by other particles (Ferrag and Maglaras, 2020). As illustrated in Figure 5, the modified PSO schema consumes fewer resources than the conventional optimization approach. It is capable of searching for enormous spaces of possible solutions (Ferrag et al., 2019a). Because it does not rely on the gradient of the issue to be optimized, as do traditional optimization methods, the problem doesn't need to be differentiable (Aslahi-Shahri et al., 2016).

The PSO is an evolutionary classifier founded on the predatory behavior of birds (Kevric et al., 2017). As a result, identifying food for birds can be comparable to determining the optimal particle solution. There are two parameters in every particle: a position variable indicated by $X_u^j = [X_{u1}^j, X_{u2}^j, X_{u3}^j, \ldots, X^j{}_{ud}]$ and a velocity parameter indicated by $V_u^j = [V_{u1}^j, V_{u2}^j, V_{u3}^j, \ldots, V_{ud}^j]$. Throughout the iteration, each particle's velocity and location update formulas are as follows:

$$V_u^j = wv_{ud}^j + e_1 n_1(pbest_{ud} - -X_{ud}^j) + e_2 n_2(gbest_d - -X_{ud}^j) \quad (5)$$

$$X_{ud}^{j+1} = X_{ud}^j + V_{ud}^{j+1} \quad (6)$$

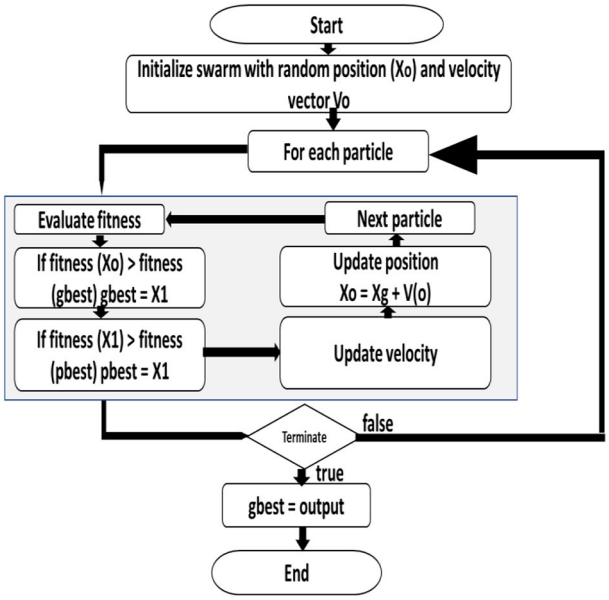

FIGURE 5
PSO algorithm schema.

TABLE 2 Position Parameters range of value.

Location	Hyperparameters used	Particle range
X^{j}_{u1}	E_1-n- filter	100-600
X^{j}_{u2}	E_1-filter-length	1-5
X^{j}_{u3}	E_1-act	Tanh(1), sigmoid (0), relu(2)
X^{j}_{u4}	E_1-G_2-droupout	0.4-0.8
X^{j}_{u5}	G_2 -neuron	256-1024
X^{j}_{u6}	G_2 -act	Tanh(1), sigmoid (0), relu(2)
X^{j}_{u7}	G_3-neuron	256-1024
X^{j}_{u8}	G_3 – act	Tanh(1), sigmoid (0), relu(2)
X^{j}_{u9}	Size-of batch	16-300
X^{j}_{u10}	Rate-of learning	0.01-1

where V^{j+1}_{ud} indicates the *d*-th constituent of the speed of the *u*-th fragment in the $j + 1$ repetition, V^{j+1}_{ud} indicates the *d*-th constituent of the speed of the *u*-th particle in the j repetition, $pbest_{ud}$ indicates the local optimal location of the *i*-th particle in the present iteration, $gbest_d$ indicates the global best location of all particles among the populace, w indicates the weight of inertia, j means the present iteration, e_1 and e_2 are acceleration coefficients referred to as social and cognitive characteristics, correspondingly, n_1 and n_2 are consecutive random numbers dispersed consistently throughout the range [0,1], X^{j+1}_{ud} indicates the *d*-th constituent of the location of the *u*-th fragment in the j+1 repetition, X^{j}_{ud} indicates the *d*-th element of the *u*-th particle's velocity in the j repetition.

TABLE 3 Searching algorithm for the optimal hyperparameters.

Algorithm	Searching for optimum hyperparameters
1.	Start;
2.	Fit the location parameter K_{xdu}
3.	Fit the rate parameter V_{xdu}
4.	While j < maximum iteration do
5.	Compute the fitness value, determine the G_{cur}, G_{avg}, and G_{min};
6.	Improve the global optimal position $fbest_d$ and local optimal position $qbest_{ud}$ concerning the value of the fitness;
7.	Improve the inertia w weight;
8.	Improve the velocity $V_{xd}{}^{u+1}$ and position K^{u+1}_{xd}
9.	u = u+1;
10.	End while
11.	Output: optimal hyperparameters

4. Proposed HAEMPSO-DNN model

The HAEMPSO algorithm is used in this research to enhance the parameter framework of a one-dimensional DNN and to determine the suitable hyperparameters, avoiding the high labor cost associated with physically changing the parameters used to discover the detection process appropriate for the network attack scenario. To begin, every layer of the one-dimensional DNN's parameters is formed of particle position parameters. Additionally, the position parameters' components for each dimension are initialized. Tables 2, 3 illustrate the parameter setting range for particle swarms.

Here, e_1- *n*-filter is the kernel number in the E_1 layer, E_1-length-filter is the filter length in the E_1 layer, E_1-act is the activation function found in the E_1 band, E1-F2- The probability of nodes remaining functional between the E_1 and G_2 layers is called dropout, the number of neurons in the G2 layer is denoted by the term "G2 neuron", G_2-act is the activation function type in G_2 layer, the number of neurons in the G3 layer is called the G3 neuron, G_3-act is the activation function in G_3 layer, the batch-size parameter specifies the size of the batch sample of training and rate-of learning is the phase to improve the weights oppositely.

4.1. Deep neural network

A DNN is a collection of multilayer perceptron's (MLPs) with a layer count >3. MLPs are a type of FFANN that are referred through the *n* layers that comprise them and benefit one another, as illustrated in Figure 6. The layer Yϵ [1, Z] of a DNN is described by D_Y (a_y, α_y, n_y). $a_y \epsilon Z$ is the neurons number in the layer. $\alpha^{ay-1}_{Y:R} \rightarrow S^{ay}$ is the transformation affine describe through the matrix W_Y and the vector $c_m.n_m : S^{ay} \rightarrow S^{ay}$ is the function transfer of the layer Y.

The matrix W_y is referred to the matrix weight between the *Y* – 1 layer and layer Y. The vector c_y is referred to as the vector bias of the Y layer. Figure 6 and Table 4 (Liu et al., 2017) reveal DNN located on MLP.

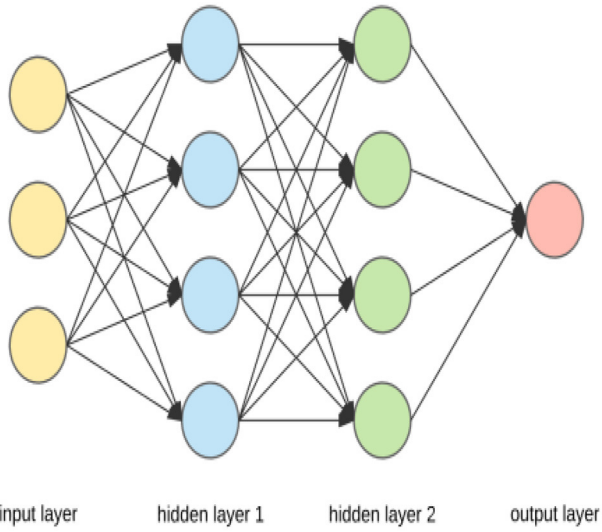

FIGURE 6
Deep neural network.

TABLE 4 DNN based on multilayer perceptron.

Algorithm	DNN grounded on multilayer perceptron
1:	Select a training pair (k, c);
2:	$h_0 = k$;
3:	For Y = 1 to Z do
4:	$g_Y = n_y\ (h_y\text{-}1) = W_{Y-1} + c_Y$;
5:	$h_Y = \alpha_M\ (g_M)$
6:	end for

TABLE 5 Parameters of DNN.

Hyperparameter	Rate	Hyperparameter	Rate
Learning rate	0.01	Input units	85
Size of the batch	32	Dropout	0.1
Training epochs	10	Activation Function	Tanh
Layers number	5	Optimizer	Stochastic gradient
Units of Hidden layer	127, 63, 32		

TABLE 6 A subset of the UNSW-NB15 dataset distribution.

Category	Train set	Test set
Benign	56,000	37,000
Backdoor	1,746	583
Exploits	33,393	11,132
Generic	40,000	18,871
Shellcode	1,133	378
Analysis	2,000	677
DoS	12,264	4,089
Fuzzers	18,184	6,062
Worms	130	44
Reconnaissance	10,491	3,496
Total	175,341	82,332

TABLE 7 Attack distribution in Bot-IoT data.

Category	Type of Attack	Train	Test
Benign	Benign	7,634	1,909
Information	OS Fingerprinting	28,662	7,166
Gathering	Service scanning	117,069	29,267
DoS	DoS TCP	985,280	246,320
Attack	DoS HTTP	2,376	594
	DoS UDP	1,652,759	413,190
Information	Data theft	94	24
Theft	Keylogging	1,175	294
DDoS	DDoS UDP	1,517,208	379,302
Attack	DDoS TCP	1,563,808	390,952
	DDoS HTTP	1,582	395
Total	-	5,877,647	1,469,413

5. Results and discussion

The hyperparameter setting of the DNN is given in Table 5 in terms of the learning rate, batch size, layers number, training epochs, input units, dropout, hidden layer units, activation, and the optimizer.

5.1. Experimentation

For the experiments, we employed two new real-world traffic databases, specifically the UNSW-NB15 data (Moustafa and Slay, 2016), and the Bot-IoT data (Shafiq et al., 2021). The statistics on threats in the Train and Testing sets in both datasets are summarized in Tables 6, 7. The experiment is carried out on Google Colaboratory using TensorFlow and the Graphics Processing Unit (GPU).

The process of constructing the training/testing sets from the UNSW-NB15 data set is illustrated in Table 6; a portion of the data set entries has been separated into training/testing sets at a ratio of around 60%:40%. To ensure the validity of NIDS evaluations, there should be no duplicate entries in the training/test sets.

Table 7 depicts the distribution of attacks that is available in the BoT-IoT dataset. The category of attacks is stated in the first column, and attack types in the second column, with the training size and testing size for each of the attack types.

TABLE 8 Confusion matrix.

	Negative class	Positive class
Class positive	False negative (FN)	True positive (TN)
Class negative	True negative (TN)	False positive (FP)

TABLE 9 Performance of the DNN model relative to the different attack types and benign in terms of DR, accuracy, and training time on the BoT-IoT dataset.

Type of Attack	DR	Accuracy	Training Time
Benign	97.92	97.54	57.5
OS Fingerprinting	97.14	97.65	67.6
Service scanning	97.53	97.65	89.1
DoS TCP	97.31	97.61	89.1
DoS HTTP	97.79	97.88	101
DoS UDP	97.62	97.66	170
Data theft	100	97.86	251
Keylogging	97.86	98.92	302
DDoS UDP	97.21	98.22	600
DDoS TCP	97.72	98.50	711
DDoS HTTP	97.79	99.22	991

5.2. Performance evaluation

We evaluate key performance parameters such as detection rate (DR), and accuracy (ACC). Table 8 illustrates the four probable classification errors.

$$DR = \frac{TP}{TP + FN} \tag{7}$$

$$TNR_{\text{Benign}} = \frac{TN}{TN + FP} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{9}$$

Where FN, TN, TP, FP, and FN mean false negative, TN means true negative, TP denotes true positive, FP connotes false positive, and TN denotes true negative, respectively.

5.3. Results

Table 9 indicates the performance of the DNN model relative to the benign and different types of attacks using the BoT-IoT dataset. It shows that the proposed HAEPSO-DNN on DoS data theft showed a DR of 100%, an accuracy of 97.86%, and a training time of 251. The OS fingerprinting showed a DR of 97.14%, an accuracy of 97.65%, and a training time of 67.6. Service scanning gave a DR of 97.53%, an accuracy of 97.65%, and a training time of 89.1. DoS TCP gave a DR of 97.31%, an accuracy of 97.61%, and a training time of 89.1. The DoS TCP gave a DR of 97.31%, an accuracy of 97.61%, and a training time of 89.1. DoS HTTP revealed an accuracy of 97.79%, an accuracy of 97.88%, and a training time of 101. The DoS UDP gave a DR of 97.62%, and an accuracy of 97.66. The Data theft gave a DR of 100, an accuracy of 97.86%, and a training time of 251. The keylogging gave a DR of 97.86%, an accuracy of 98.92%, and a training time of 302 ms. The DDoS UDP gave a DR of 97.21%, an accuracy of 98.22%, and a training time of 600. The DDoS TCP gave a DR of 97.72%, an accuracy of 98.50%, and a training time of 711 s. The DDoS HTTP gave a DR of 97.79%, an accuracy of 99.22%, and a training time of 991 s. As seen in Table 9, the data theft attack gave the highest DR at a relatively high training time of 251 s. Whereas, the keylogging attack gave a high accuracy of 98.92% at the expense of the training time of 302 s.

TABLE 10 Performance of the DNN model relative to the different attack types and benign in terms of DR, accuracy, and training time on the UNSW-NB15 dataset.

Type of attack	DR	Accuracy	Training time
Benign	99.7	99.9	70.7
DoS	98.3	98.5	92.6
Back door	96.2	97.3	102
Worm	82.7	83.8	92
Shellcode	91.2	92.2	120
Probe	91.3	93.4	82
Exploits	99.0	98.10	31
Fuzzer	88.1	89.2	77
Analysis	93.3	94.4	51
Generic	99.0	99.9	66
Reconnaissance	93.0	94.1	41

The performance of the DNN model against benign and different types of attacks is shown in Table 10 using the UNSW-NB15 dataset. It demonstrates that the proposed HAEPSO-DNN on Benign achieved a DR of 99.7%, an accuracy of 99.9%, and a training time of 70 ms. The DoS gave a DR of 98.3%, an accuracy of 98.5%, and a training time of 92.6%. Backdoor had a detection rate of 97.14%, an accuracy of 97.65%, and a training time of 67 s. The Worm resulted in a DR of 82.7%, an accuracy of 83.8%, and 92% training time. The shellcode had a DR of 91.2%, an accuracy of 92.2%, and a training time of 120 s. The probe yielded a DR of 91.3%, an accuracy of 92.2%, and a training time of 82 s. The exploits gave a DR of 99%, an accuracy of 98.10%, and a training time of 31 s. The fuzzer gave a DR of 88.1%, an accuracy of 89.2%, and a training time of 77 s. The analysis revealed a DR of 93.3%, an accuracy of 94.4%, and a training time of 51 s. The Generic gave a DR of 99.9%, an accuracy of 98.8%, and a training time of 66 s. The reconnaissance gave a DR of 93%, an accuracy of 94.1% s, and a training time of 41 s. The findings showed that the generic attack gave the highest DR reaching 99.9%, accuracy of 98.8%, and training time of 66 s out of all the attack classes. The benign class gave the highest accuracy of 99.9%, DR of 99.7%, and training time of 70.7 s.

The performance of the DNN approach in terms of DR, training time, and accuracy on BoT-IoT and UNSW-NB15 datasets is shown in Figure 7.

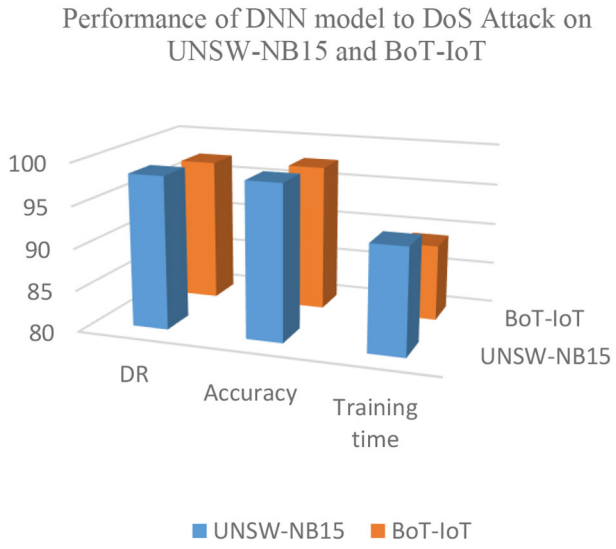

FIGURE 7
Performance of HAEPSO-DNN model concerning DOS attack on the UNSW-NB15 and BoT-IoT datasets.

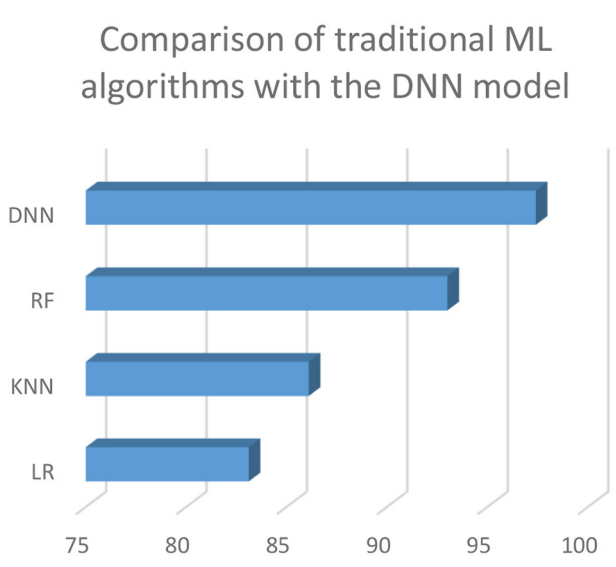

FIGURE 8
Comparison with the traditional ML algorithms.

5.4. Comparison with the machine learning model

We compare our suggested DNN technique with traditional ML classifiers. As shown in Figure 8, we employed a 10-fold cross-validation (CV) technique to test the efficiency of the RF, LR, KNN, and DNN model. As can be seen from the data, KNN has an accuracy of 86.11, 93.02% for RF, 83.12% for the LR model, and 97.41% for the DNN classifier on the UNSW-NB15 dataset. The LR classification algorithm does not work well since its performance is dependent on the feature design of the raw data. While RF produces higher accuracy than a linear model since it can learn more through the bagging of several decision trees, it is unable to extract deep information from the training data. As a result, the DNN model defeats it.

TABLE 11 Comparison with recent studies.

Authors	Dataset	Accuracy	DR
Ravi et al. (2022)	UNSW-NB15	99%	x
Chohra et al. (2022)	UNSW-NB15	89.52%	x
Proposed	UNSW-NB15	99.9%	99%

5.5. Comparison with the existing DL models

As seen in Table 11, we compared the performance of our proposed model in this section with the recent work (Chohra et al., 2022; Ravi et al., 2022). Ravi et al. (2022) proposed a DL model for attack detection in CPS. The simulation findings gave an accuracy of 99%, precision of 99%, recall of 99%, and f1-score of 99%. However, Ravi et al. (2022) did not pay attention to DR which is an important metric in IDS. Also, Ravi et al. (2022) considered the problem as a multi-class classification problem. Chohra et al. (2022) adopted PSO for feature selection and used an ensemble model for classification. The experiment was performed on the UNSW-NB15 dataset. The findings gave an accuracy of 89.52% on a binary classification problem. In our proposed model, our model gave an accuracy of 99.9%, DR of 99%, and training time of 66 s. Additionally, our model addressed the problem as a multi-class problem that outperformed the multi-class problem of Ravi et al. (2022), and binary problem as suggested by Chohra et al. (2022) in terms of accuracy and DR.

6. Threats to validity

This study utilized two public datasets from distinct sources that are suitable for an IoT setting to evaluate the performance of the suggested solution. Due to the use of open-source assessment and classification tools, the results cannot be generalized to closed-source techniques or proprietary data, which may result in performance variation.

Also, 10-fold cross-validation was used as validation to eliminate bias in the proposed work. Other validation methods (such as *K*-fold or five-fold) may not yield the same results. Different splits rate can also result in varied performances.

7. Conclusion and future work

This research proposes a new HAEPSO-DNN algorithm and successfully applies it to the detection of multiple types of intrusions in IoT networks. However, the inertia weight component is adaptively modified in response to the fitness value to avoid PSO entering the local extremum problem and obtaining the appropriate DNN overall parameters. The experimental

analysis was performed on two realistic UNSW-NB15 and BoT-IoT datasets that are well suited for the IoT ecosystem as against outdated datasets used in previous research. Additionally, the suggested approach reduces communication overhead and eliminates the necessity for a foreign key, which is essential for IoT network security encryption solutions. Our findings indicate that the proposed technique delivers greater accuracy, with DR indicating a more robust detection performance. Based on experimental findings from network and testbed simulations, we can conclude that implementing the proposed HAEPSO-DNN algorithms for successful NIDS in the IoT ecosystem is both practicable and feasible. The infrastructure can be organized in such a way that it detects threats in the IoT ecosystem as a unit including healthcare devices and smart homes. The future study will involve experimenting with more DL models for noticing IoT network intrusions, including CNN, LSTM, and recurrent neural networks.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://research.unsw.edu.au/projects/bot-iot-dataset.

Author contributions

YS: formal analysis, investigation, writing review, and coding. AU: methodology, resources, writing—review and editing, and software. FS: literature analysis and data collection. MA: algorithm development, coding, and validation. All authors contributed to the article and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abbas, N., Asim, M., Tariq, N., Baker, T., and Abbas, S. (2019). A mechanism for securing IoT-enabled applications at the fog layer. *J. Sens. Actuator Netw.* 8, 1–18. doi: 10.3390/jsan8010016

Ahanger, T. A., Aljumah, A., and Atiquzzaman, M. (2022). State-of-the-art survey of artificial intelligent techniques for IoT security. *Comput. Netw.* 19, 108771. doi: 10.1016/j.comnet.2022.108771

Alterazi, H. A, Kshirsagar, P. R, Manoharan, H, Selvarajan, S, Alhebaishi, N, Srivastava, G, et al. (2022). Prevention of cyber security with the internet of things using particle swarm optimization. *Sensors* 22, 6117. doi: 10.3390/s22166117

Arshad, J., Azad, M. A., Abdeltaif, M. M., and Salah, K. (2020). An intrusion detection framework for energy constrained IoT devices. *Mech. Syst. Signal Process.* 136, 106436. doi: 10.1016/j.ymssp.2019.106436

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* 169, 1–12. doi: 10.1016/j.compstruc.2016.03.001

Aslahi-Shahri, B. M., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M. J. et al. (2016). A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Comput. Appl.* 27, 1669–1676. doi: 10.1007/s00521-015-1964-2

Atlam, H. F., Alenezi, A., Alassafi, M. O., Alshdadi, A. A., and Wills, G. B. (2022). "Security, cybercrime and digital forensics for IoT," in *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm. Intelligent Systems Reference Library*, eds. S. Peng, S. Pal, and L. Huang (Berlin: Springer, Cham).

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: a survey. *Comput. Netw.* 54, 2787–2805. doi: 10.1016/j.comnet.2010.05.010

Aydin, M. A., Zaim, A. H., and Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. *Comput. Electr. Eng.* 35, 517–526. doi: 10.1016/j.compeleceng.2008.12.005

Blanco, R., Malagón, P., Briongos, S., and Moya, J. M. (2019). "Anomaly detection using gaussian mixture probability model to implement intrusion detection system," in *Hybrid Artificial Intelligent Systems. HAIS 2019. Lecture Notes in Computer Science*, eds. G. H. Pérez, G. L. Sánchez, L. M. Castejón, H. Quintián, and R. E. Corchado (Berlin, Germany: Springer), 648–59.

Chohra, A., Shirani, P., Karbab, E. B., and Debbabi, M. (2022). CHAMELEON: Optimized feature selection using particle swarm optimization and ensemble methods for network anomaly detection. *Comput. Secur.* 117, 102684. doi: 10.1016/j.cose.2022.102684

Choudhary, S., and Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT. *Procedia Comput. Sci.* 167, 1561–1573. doi: 10.1016/j.procs.2020.03.367

Chung, Y. Y., and Wahid, N. (2012). A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Appl. Soft Comput. J.* 12, 3014–3022. doi: 10.1016/j.asoc.2012.04.020

Derhab, A, Guerroumi, M, Gumaei, A, Maglaras, L, Ferrag, M. A, Mukherjee, M, et al. (2019). Blockchain and Random Subspace Learning-Based IDS for SDN-Enabled Industrial IoT Security. *Sensors (Switzerland)* 19, 1–24. doi: 10.3390/s19143119

Elbasiony, R. M., Sallam, E. A., Eltobely, T. E., and Fahmy, M. M. (2013). A hybrid network intrusion detection framework based on random forests and weighted *k*-means. *Ain Shams Eng. J.* 4, 753–762. doi: 10.1016/j.asej.2013.01.003

Fenanir, S., Semchedine, F., and Baadache, A. (2019). A Machine Learning-Based Lightweight Intrusion Detection System for the Internet of Things. *Rev. d'Intelligence Artif.* 33, 203–211. doi: 10.18280/ria.330306

Ferrag, M. A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., Janicke, H., et al. (2019b). Blockchain technologies for the internet of things: research issues and challenges. *IEEE Internet Things J.* 6, 2188–2204. doi: 10.1109/JIOT.2018.2882794

Ferrag, M. A., and Maglaras, L. (2019). Deliverycoin: An IDS and blockchain-based delivery framework for drone-delivered services. *Computers* 8, 1–15. doi: 10.3390/computers8030058

Ferrag, M. A., and Maglaras, L. (2020). DeepCoin: a novel deep learning and blockchain-based energy exchange framework for smart grids. *IEEE Trans. Eng. Manag.* 67, 1285–1297. doi: 10.1109/TEM.2019.2922936

Ferrag, M. A., Maglaras, L., Ahmim, A., Derdour, M., and Janicke, H. (2020). RDTIDS: Rules and decision tree-based intrusion detection system for internet-of-things networks. *Futur. Internet* 12, 1–14. doi: 10.3390/fi12030044

Ferrag, M. A., Maglaras, L. A., Janicke, H., and Smith, R. (2019a). Deep Learning Techniques for Cyber Security Intrusion Detection : A Detailed *Analysis*. 126–136. doi: 10.14236/ewic/icscsr19.16

Govindarajan, M., and Chandrasekaran, R. (2011). Intrusion detection using neural based hybrid classification methods. *Comput. Netw.* 55, 1662–1671. doi: 10.1016/j.comnet.2010.12.008

Habib, M., Aljarah, I., and Faris, H. (2020). A modified multi-objective particle swarm optimizer-based lévy flight: an approach toward intrusion detection in internet of things. *Arab. J. Sci. Eng.* 45, 6081–6108. doi: 10.1007/s13369-020-04476-9

Kayode Saheed, Y., Idris Abiodun, A., Misra, S., Kristiansen Holone, M., and Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Eng. J.* 61, 9395–9409. doi: 10.1016/j.aej.2022.02.063

Kevric, J., Jukic, S., and Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Comput. Appl.* 28, 1051–1058. doi: 10.1007/s00521-016-2418-1

Kim, G., Lee, S., and Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst. Appl.* 41, 1690–1700. doi: 10.1016/j.eswa.2013.08.066

Lahasan, B., and Samma, H. (2022). Optimized deep autoencoder model for internet of things intruder detection. *IEEE Access* 10, 8434–8448. doi: 10.1109/ACCESS.2022.3144208

Leo, M., Battisti, F., Carli, M., and Neri, A. (2014). A federated architecture approach for Internet of Things security. *2014 Euro Med Telco Conference From Netw. Infrastructures to Netw. Fabr. Revolut. Edges, EMTC 2014*. Piscataway, NJ: IEEE

Li, S., Li, Y., Han, W., Du, X., Guizani, M., Tian, Z., et al. (2021b). Malicious mining code detection based on ensemble learning in cloud computing environment. *Simul. Model. Pract. Theory* 113, 102391. doi: 10.1016/j.simpat.2021.102391

Li, S., Zhang, Q., Wu, X., Han, W., and Tian, Z., Attribution classification method of APT malware in IoT Using Machine Learning Techniques. *Secur. Commun. Netw.* (2021a) 2021, 1–12. doi: 10.1155/2021/9396141

Li, W., Meng, W., and Au, M. H. (2020). Enhancing collaborative intrusion detection *via* disagreement-based semi-supervised learning in IoT environments. *J. Netw. Comput. Appl.* 161, 1–9. doi: 10.1016/j.jnca.2020.102631

Liang, C., Shanmugam, B., Azam, S., Karim, A., Islam, A., Zamani, M., et al. (2020). Intrusion detection system for the internet of things based on blockchain and multi-agent systems. *Electron.* 9, 1–27. doi: 10.3390/electronics9071120

Lin, W. C., Ke, S. W., and Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowl. Based Syst.* 78, 13–21. doi: 10.1016/j.knosys.2015.01.009

Liu, J., Yang, D., Lian, M., and Li, M. (2021). Research on intrusion detection based on particle swarm optimization in IoT. *IEEE Access* 9, 38254–38268. doi: 10.1109/ACCESS.2021.3063671

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F. E., et al. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing* 234, 11–26. doi: 10.1016/j.neucom.2016.12.038

Marlow, R, Kuriyakose, S, Mesaros, N, Han, H. H, Tomlinson, R, Faust, S. N, et al. (2018). A phase III, open-label, randomised multicentre study to evaluate the immunogenicity and safety of a booster dose of two different reduced antigen diphtheria-tetanus-acellular pertussis-polio vaccines, when co-administered with measles-mumps-rubella vacci. *Vaccine* 36, 2300–2306. doi: 10.1016/j.vaccine.2018.03.021

Minh Dang, L., Piran, M. J., Han, D., Min, K., and Moon, H. (2019). A survey on internet of things and cloud computing for healthcare. *Electron.* 8, 1–49. doi: 10.3390/electronics8070768

Moustafa, N., and Slay, J. (2016). The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J.* 25, 18–31. doi: 10.1080/19393555.2015.1125974

Oh, D., Kim, D., and Ro, W. W. (2014). A malicious pattern detection engine for embedded security systems in the internet of things. *Sensors (Switzerland)* 14, 24188–24211. doi: 10.3390/s141224188

Pongle, P., and Chavan, G. (2015). Real time intrusion and wormhole attack detection in internet of things. *Int. J. Comput. Appl.* 121, 1–9. doi: 10.5120/21565-4589

Putra, G. D., Dedeoglu, V., Kanhere, S. S., and Jurdak, R. (2020). Poster abstract: Towards scalable and trustworthy decentralized collaborative intrusion detection system for IoT. *Proc. - 5th ACM/IEEE Conf. Internet Things Des. Implementation, IoTDI* 2020, 256–257. doi: 10.1109/IoTDI49375.2020.00035

Ramadan, R. A., and Yadav, K. (2020). A novel hybrid intrusion detection system (Ids) for the detection of internet of things (IoT) network attacks. *Ann. Emerg. Technol. Comput.* 4, 61–74. doi: 10.33166/AETiC.2020.05.004

Ravi, V., Chaganti, R., and Alazab, M. (2022). Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Comput. Electr. Eng.* 102, 108156. doi: 10.1016/j.compeleceng.2022.108156

Raza, S., Wallgren, L., and Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Networks* 11, 2661–2674. doi: 10.1016/j.adhoc.2013.04.014

Saheed, Y. K. (2022a). "Performance improvement of intrusion detection system for detecting attacks on internet of things and edge of things," in *Artificial Intelligence for Cloud and Edge Computing. Internet of Things*, eds. S. Misra, A. Kumar Tyagi, V. Piuri, and L. Garg (eds). Berlin, Germany: Springer, Cham.

Saheed, Y. K. (2022b). "A binary firefly algorithm based feature selection method on high dimensional intrusion detection data," in *Illumination of Artificial Intelligence in Cybersecurity and Forensics. Lecture Notes on Data Engineering and Communications Technologies*, eds. S. Misra and C. Arumugam. Berlin, Germany: Springer, Cham.

Saheed, Y. K., Arowolo, M. O., and Tosho, A. U. (2022b). An efficient hybridization of *K*-means and genetic algorithm based on support vector machine for cyber intrusion detection system. *Int. J. Electr. Eng. Inform.* 14, 426–442. doi: 10.15676/ijeei.2022.14.2.11

Saheed, Y. K., Baba, U. A., and Raji, M. A. (2022a). "Big data analytics for credit card fraud detection using supervised machine learning models," in *Big Data Analytics in the Insurance Market (Emerald Studies in Finance, Insurance, and Risk Management)*, eds. K. Sood, B. Balusamy, S. Grima, and Marano. Bingley, United Kingdom: Emerald Publishing Limited, 31–56.

Saheed, Y. K., and Hamza-usman, F. M. (2020). Feature Selection with IG-R for Improving Performance of Intrusion Detection System. *Int. J. Commun. Netw. Inform. Secur.* 12, 338–344. doi: 10.17762/ijcnis.v12i3.4569

Sedjelmaci, H., Senouci, S. M., and Al-Bahri, M. (2016). A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology. *2016 IEEE Int. Conf. Commun. ICC* 2016. Bangalore, India: HAL.

Shafiq, M., Tian, Z., Bashir, A. K., Du, X., and Guizani, M. (2021). CorrAUC: A Malicious Bot-IoT traffic detection method in IoT network using machine-learning techniques. *IEEE Internet Things J.* 8, 3242–3254. doi: 10.1109/JIOT.2020.3002255

Shafiq, M., Tian, Z., Sun, Y., Du, X., and Guizani, M. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. *Futur. Gener. Comput. Syst.* 107, 433–442. doi: 10.1016/j.future.2020.02.017

Sicari, S., Rizzardi, A., and Coen-Porisini, A. (2020). 5G In the internet of things era: An overview on security and privacy challenges. *Comput. Networks* 179, 107345. doi: 10.1016/j.comnet.2020.107345

Singh, S., Sharma, K., Yoon, B., Shojafar, M., Cho, G. H., Ra, I. H., et al. (2020). Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city. *Sustain. Cities Soc.* 63, 102364. doi: 10.1016/j.scs.2020.102364

Subham, K. G., Meenakshi, T., and Jyoti, G. (2022). Hybrid optimization and deep learning based intrusion detection system. *Comput. Electr. Eng.* 100, 1–15. doi: 10.1016/j.compeleceng.2022.107876

Thamilarasu, G., and Chawla, S. (2019). Towards deep-learning-driven intrusion detection for the internet of things. *Sensors (Switzerland)* 19, 1977. doi: 10.3390/s19091977

Thanigaivelan, N. K., Nigussie, E., Kanth, R. K., Virtanen, S., and Isoaho, J. (2016). Distributed internal anomaly detection system for Internet-of-Things. *2016 13th IEEE Annu. Consum. Commun. Netw. Conf. CCNC 2016*, 319–320.

Wang, G., Hao, J., Mab, J., and Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* 37, 6225–6232. doi: 10.1016/j.eswa.2010.02.102

Yan, B., and Han, G. (2018). Effective feature extraction *via* stacked sparse autoencoder to improve intrusion detection system. *IEEE Access* 6, 41238–41248. doi: 10.1109/ACCESS.2018.2858277

Zhang, W., and Zhang, Y. (2022). Intrusion detection model for industrial internet of things based on improved autoencoder. *Comput. Intell. Neurosci.* 27, 2022. doi: 10.1155/2022/1406214