Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 3939895, 15 pages
https://doi.org/10.1155/2023/3939895

Research Article

Industrial Internet of Things Intrusion Detection Method Using Machine Learning and Optimization Techniques

Tarek Gaber,[1,2] **Joseph B. Awotunde**,[3] **Sakinat O. Folorunso**,[4] **Sunday A. Ajagbe**,[5] **and Esraa Eldesouky**[6,7]

[1]*Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt*
[2]*School of Science, Engineering, and Environment, University of Salford, Manchester M5 4WT, UK*
[3]*Department of Computer Science, Faculty of Information and Communication Sciences, University of Ilorin, Ilorin 240003, Nigeria*
[4]*Department of Mathematical Science Olabisi Onabanjo University, Ago-Iwoye, 120107, Nigeria*
[5]*Department of Computer & Industrial Production Engineering, First Technical University, Ibadan, 200255, Nigeria*
[6]*Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia*
[7]*Department of Computer Science, Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt*

Correspondence should be addressed to Tarek Gaber; t.m.a.gaber@salford.ac.uk

Received 9 April 2022; Revised 29 August 2022; Accepted 30 September 2022; Published 30 April 2023

Academic Editor: Basem M. Elhalawany

The emergence of the Internet of Things (IoT) has witnessed immense growth globally with the use of various devices found in home, transportation, healthcare, and industry. The deployment and implementation of the IoT paradigm in industrial settings lead to the architectural changes of Industrial Automation and Control Systems (IACS) plus the countless connectivity of industrial systems. This resulted in what is referred to as the Industrial Internet of Things (IIoT), which removes the barrier of connecting IACS to isolated conventional ICT platforms. In recent times, the IoT has started hacking our personal lives and not only our world, thus creating a platform for impending IoT cyberattacks. The widespread use of the IoT has created a rich platform for possible IoT cyberattacks. Machine learning (ML) algorithms have been driven solutions to secure wireless communication in IIoT-based systems, and their use in solving various cybersecurity challenges. Therefore, this paper proposes a novel intrusion detection model based on the Particle Swarm Optimization (PSO) and Bat algorithm (BA) for feature selection, and the Random Forest (RF) classifier for the classification of malicious behaviors in IIoT-based network traffic. An IIoT-based cybersecurity dataset, WUSTL-IIOT-2021 Dataset, was used to evaluate the performance of the proposed model using accuracy, recall, precision, and F1-score. The results of the two feature selection were compared to identify the most promising one. The results were compared with other recent state-of-the-art ML and multiobjective algorithms, and the results showed better performance. The RF along with BA classifier had proved to be the best classifier.

1. Introduction

The emergence of the Internet of Things (IoT) paradigm in Industrial Automation and Control Systems (IACS) is termed the Industrial Internet of Things (IIoT), and in recent years it has become very popular. The IACS have been utilized in recent time to keep an eye on industrial machines and processes, and thus the IIoT-based systems have become an essential part of every critical infrastructure in smart industries. The largest parts of these systems are the data acquisition and supervisory systems that repeatedly manage the IACSs. Real-time monitoring, interaction with the devices, analysing of data, and logging all the events that happen in the systems are the main roles of these systems. Hence, the arrival of the IoT paradigm in these systems enriches the security and network intelligence in the computerization and optimization of

industrial processes. Since the operations of IIoTs lead to a huge amount of data, and the majority of the applications are mission-critical and demand high availability, there is need for cyber-security to properly secure these systems.

Isolating IACSs from the outside world in the past has really helped to secure IACSs from intrusion and malicious external attack [1]. The recent improvements and usage of Internet communication with increased connectivity to transmit information have created more avenues for cyber-attacks like Denial-of-Service attack, Man-in-the-Middle (MITM) attack, Phishing Attack, Password Attack, SQL Injection Attack, and Cryptojacking against these systems [2, 3]. Cyber-attacks have a number of detrimental repercussions. When an attack is attempted, it may result in data breaches, which may cause data loss or manipulation. Companies suffer financial losses, a decrease in customer trust, and damaged reputations. In order to prevent or stop cyber-attacks, a cyber-security can use measures like IDS and antivirus to preventing unwanted digital access to networks, computer systems, and their parts. Hence, security is the most concerning issue in IIoT-based systems due to the sensitive nature of the industrial application.

To provide a secure environment, an intrusion detection system (IDS) has been an integral part of IIoT-based applications since the intrusion of crucial security concerns in 2010, the Stuxnet worm was exposed [4], and in December 2017, the attack reappeared with another powerful malware called Triton against the IACSs [5]. These attacks give rise to the awareness of the necessity to pay attention to the protection of these vital infrastructures' security [3]. The fundamental difference between regular information technology systems, and the IACSs necessities their priorities to secure common vulnerabilities, and in most cases their attacks are different [6]. Additionally, IACS traffic and data type are specifically different using certain IIoT communication protocols like Distributed Network Protocol 3 (DNP3), Building Automation Controls Network (BACnet), and Modbus [7]. Hence, with these special reasons, the security of IIoT-based applications must be properly considered when it comes to the designing of an IDS for IACSs.

The continuous growth of IoT-based systems and their related applications demands the improvement of network security and to maintain the security of any interconnected system that requires protection of its integrity, availability, and confidentiality [8]. The most common IIoT-based system threats that interrupt and attempt to terminate the integrity, availability, and/or confidentiality are cybersecurity and intrusions. IDS applications include the hardware devices or software services that monitor the network for malicious activities. The network intrusion detection system (NIDS) plays a prominent role in addressing various Internet attacks, and the IIoT has been identified as an integral part of the present machinery for industrial data transfer, necessitating the need for network security. The NIDS are used to safeguard the workstation structures from network intrusion and multiple grid invasions. Recent work has created new IDSs in response to the attacks and threats posed by various aggressive frameworks. However, the performance of current machine learning-based methods in terms of accuracy and high false alarm rate are still issues that need urgent attention in order to reduce the irregularity of discovery methods of intrusion and malicious attacks.

Recently, feature selection has been identified as a modern method of getting an accurate and low false alarm rate in NIDSs [9, 10]. This method is used to select the most useful and fit features for a better classification result in NIDS models. This led to aggregated accuracy performance and a reduced error rate in their applications for detection of attackers [9]. Additionally, the datasets of features are very huge, and not all are always useful for the classification of the dataset as either normal or abnormal. Hence, the use of feature selection techniques is very necessary. The use of feature selection is, therefore, very important in the use of NIDSs in IIoT-based network traffic, helping in getting optimal results from the classification models used.

ML-based models have been previously used in securing IT-based systems [11–13] and IoT-based networks, but the suitability of these models has not been widely employed and still remains debatable according to authors in [12]. The ability to detect any penetration into the system is the main security concern of the IIoT-based devices. Sometimes the ML-based models of IDSs for IACS may not be able to properly detect the attack due to its design, which may not address the imbalance of the data, which is the main property of intrusion detection problems [14]. Hence, for better IDSs performance, the issue of imbalanced datasets for IIoT-based systems should be considered, addressing the questions of what are the true boundaries and how do various performance metrics react to them?

The application of the ML-based model for the IIoT-based network still faces various challenges, as given in the following:

(i) *The Issue of Low Processing Ability*. The IoT-based devices have energy constraints with limited processing capacity due to their small size. This creates huge challenges since ML-based models require real-time processing of data, thus their implementation in such resource-constrained environments creates issues

(ii) *Data Analytics*. Data is generated heterogeneously in the IoT environment and demands preprocessing before being applied to an ML-based model. This necessitates the processing memory space and power of IoT-based devices, making the provision of an efficient solution a challenge for diverse data

Inspired by the aforementioned challenges, the assumption of working on the imbalanced datasets by turning them into class balanced datasets, the aim of this paper is to design an efficient and yet accurate intrusion detection method for IIoT applications. Bioinspired optimizations (i.e., PSO [15] and BAT [16]) were used to get a subset of features helping to achieve this aim. Also, Random Forest (RF), k-Nearest Neighbor (k-NN), and MultiLayer Perceptron (MLP) classifiers were employed to measure the performance in terms of accuracy, precision, recall, F1-score, and ROC. Also, the experiments were conducted on a dataset (i.e., the WUSTL-IIOT-2021 Dataset) which was collected specifically for IIoT cybersecurity threats and attacks.

1.1. The Study Has the following Significant Contributions

(i) Proposing an intrusion detection method for IIoT applications using bioinspired-based feature selection to enhance the performance of the intrusion detection system through reducing the number of the selected features while getting a high accuracy

(ii) Investigating the effectiveness of the proposed feature selection method above by different types of machine learning algorithms (i.e., RF, k-NN and MLP). This was done with a relevant dataset, WUSTL-IIOT-2021 dataset, which was collected in IIoT environment

(iii) Providing a thorough evaluation through two phases: (1) using the benchmark evaluation metrics of accuracy, precision, recall, F1-score, and ROC and (2) comparing the obtained results with the most related published work which showed better results for the proposed method

The rest of the paper is organized as follows: Section 2 presents the literature review on ML-based models for intrusion detection on IIoT networks. Section 3 explains the methodology employed in this study. Section 4 presents the experimental results of the study, while Section 5 concludes the paper with future direction.

2. Related Work

The IIoT idea was created specifically for application in modern industry. Modern IIoT refers to the application of the standard IoT in various industrial projects and businesses. Numerous actuators, sensors, control systems, interfaces for communication and integration, cutting-edge security systems, networks for automobiles, household appliances, etc., are all included in the IIoT. The IIoT's nodes can all connect to the Internet. The capacities of many sectors, manufacturing facilities, asset management systems, sophisticated logistics systems, etc., have been substantially improved by the use of IIoT in contemporary businesses. Several applications, gadgets, and services can connect the real area to a virtual one thanks to the IIoT [17].

There are various ways for IIoT nodes to connect to the Internet, including through the use of Message Queue Telemetry Transport (MQTT), Modbus TCP, cellular networks, Long-Range Radio Wide Area Network (LoRaWAN), and other TCP/IP-based communication protocols [18]. The majority of IIoT nodes can also gather, process, and transfer data. Due to their capabilities, they are vulnerable to several privacy and security risks that could endanger IIoT systems and the applications they are a part of [19]. The fact that IIoT nodes are constantly active while carrying out data collecting, processing, and transmission is one of their major characteristics.

The perception layer, the network layer, the application layer, and the Cloud are the three main layers of the IIoT. These levels are founded on data flow. Additionally, each layer is vulnerable to different kinds of assaults and breaches that could jeopardize the IIoT systems. Access control breaches, data corruption incidents, spoofing assaults, Distributed DoS, Operating System (OS) attacks, and jammer attacks are some frequent attacks and intrusions on the IIoT ecosystem. Many firms are employing intrusion detection systems to prevent these malicious assaults, ensuring that IIoT networks' security and active IIoT nodes' security are maintained (IDSs). Additionally, these IDSs can be set up at any layer.

There have been various approaches to solving the problem of identifying intrusions like ML-models, ensemble methods, deep learning methods, and the hybrid approaches enabled by feature selection [20, 21]. Through the analysis of collected information, the NIDS can detect attacks from various network traffic and systems [22]. Hence, the approach is widely used as a technique for network security. Various research have used both ML and DL methods for the purpose of intrusion detection in various environments like the World Wide Web, IoT-based systems, and Internet network traffic for the purpose of detecting and categorizing attacks such as [23, 24], among others. In recent time, ML and DL techniques, like SVM, RBM [25], Conventional Neural Network (CNN) [26], Artificial Neural Network (ANN) [27], Decision Tree (DT) and Random Tree (DT) [28], and clustering and K-NN algorithms [29], have been used for improving intrusion detection systems. The advantages of the ML-based IDS model are as follows:

(i) The ML-based models can efficiently detect attacks with small variations since they are trained based on the behavior/pattern of the network for most scenarios

(ii) The use of unsupervised learning models can easily detect zero-day attacks, especially if the model is trained based on this method

(iii) Even in complex network environments, ML-based IDS gives higher detection accuracy and is faster

Machine learning approaches have been shown to provide effective intrusion detection systems during the recent years. They produce better outcomes than other alternative methods since they are applicable to different types of datasets and can analyze real-time data. Researchers usually use various approaches, including deep learning, heuristics, adaptive learning, decision trees, and semisupervised learning.

Priya et al. [30] proposed a two-phase intrusion detection model that was developed that includes SVM, NB, and DT in the first phase and an RF classifier for prediction using ensemble learning. In addition, to deliver better predictions, the results of the ANN classifier were integrated with those of the RF. The combined model is validated against the WUSTL_IIOT-2018, N_BaIoT, and Bot_IoT datasets. According to the conducted results of applying only the first phase, the Naïve Bayes classifier had the lowest accuracy, followed by the SVM and DT classifiers, while the DT classifier achieved the highest accuracy of 96%. The proposed method, on the other hand, incorporated ANN and RF predictions and attained a 99 percent accuracy rate for all the three datasets. A deep learning strategy was used to address another IIOT intrusion detection model by Raja [31]. The

proposed DL-TL-NIDS model had two levels of detection. The DNN is trained and evaluated at the first level to detect current assaults. Attacks that had a poor detection or low accuracy rate were classified as challenging attacks. These challenging attacks are input to second-level detection, which trains the Negative Selection Algorithm (NSA) and DNN models using the Dragonfly algorithm. Finally, the outputs of both models are combined using Dempster Shafer's combination rule.

Nevertheless, using bioinspired algorithms to extract key IIOT network features can assist in reducing processing costs and memory use and make it easier to apply various classification approaches to the selected features. In this section, we present related works on managing intrusion detection in the IIoT that use bioinspired algorithms for feature selection.

Keserwani et al. [32] suggested a hybrid metaheuristic approach for feature selection and deep learning for classification to identify intrusions in a virtualized cloud network. A deep sparse auto-encoder is utilized to classify the important features from the cloud network connections, which are identified using hybrid Gray Wolf Optimization (GWO) and PSO. The authors expanded on their previous work in [20] to include fetch attacks in the IoT world. The hybrid GWO-PSO is also utilized to extract key IoT network properties, which are then fed into a random forest classifier for improved attack detection accuracy. The proposed model was tested on the KDDCup99, NSL–KDD, and CICIDS-2017 datasets, and it achieved an accuracy of 99.66%.

Kasongo [19] proposed an IDS for IIOT by employing the genetic algorithm along with a random forest model, which was utilized in the fitness function of the genetic algorithm. The usage of Genetic Algorithms (GA) is motivated by the presence of a large number of features in current datasets, as well as a large number of network traces. As a result, the ML algorithms' training process is badly impacted and misled, as ML performance decreases as the number of features grows. Hence, the learning process becomes more difficult as the dataset's number of characteristics rises. Therefore, the genetic algorithm is utilized to improve the feature selection, and the author used tree-based methods such as RF, DT, and ET algorithms for each attribute vector, all of which were tested on the UNSW-NB15 general-purpose dataset.

Awotunde et al. [3] utilized the same dataset, together with the NSL-KDD dataset, to build a hybrid rule-based feature selection technique. The proposed research combines a deep feedforward neural network model and rule-based feature selection with IIOT applications to obtain relevant data that may be utilized to construct an intelligent NIDS (i.e., data gathered from TCP/IP packets). This research presents a three-tier methodology for intrusion detection in IIoT systems, in which a rule-based model is utilized for feature selection and a genetic tool is employed to create the characteristics with the highest values. Finally, the selected features are loaded into the ANN for use in the learning process.

The authors in [33] employed the Aquila optimizer (AQU) for feature selection in the CIC2017, NSL-KDD, BoT-IoT, and KDD99 datasets to assess the quality of the proposed IDS approach. A light feature extraction strategy based on CNN was adopted to extract relevant features from the datasets utilized in this work. Following that, the AQU algorithm is used to pick a group of the best features that shows the datasets properties.

The ML-based IDS has a lot of advantages, like being faster and more accurate in both simple and complex environments. Furthermore, owing to the training nature of ML models, particularly through unsupervised learning techniques, several types of assaults may be easily spotted. Yet, several challenges still remain when applying machine learning models to IIoT networks. The bulk of recent datasets are large in size, both in terms of feature space dimension and the number of network traces. The presence of a large number of features in a dataset might have a detrimental influence on the training process of machine learning algorithms. The performance of the ML-based IDS has therefore deteriorated since performing an effective learning process becomes more difficult as the number of characteristics in a dataset grows. In order to obtain the essential features, an accurate method of feature selection is required. Another issue is the lack of real-world data collected by an IIoT system in order to assess the efficacy of present solutions.

Another point, which is not well-addressed in the literature, is the imbalance of the dataset used in building ML-based intrusion detection systems. Because of the imbalanced datasets, minority attacks may be missed. Also, the IDS model can identify the majority of attacks, but due to the imbalance, certain attacks may not be detected. As a result, these attacks need a high level of detection. Table 1 shows the summary of the main findings in the reviewed literature.

The application of feature selection has been helped in the area of feature reduction to transform features from high dimensional to a lower dimensional space without reducing the efficiency of the prediction algorithms. This technique is used to eliminate irrelevant features and variables from any dataset without reducing the data's usefulness to the classification model.

From the literature review, there has not been any work that applies feature selection on WUSTL-IIOT-2021 datasets for IDS to the best of our knowledge, this study will be the first to apply feature selection for IIoT-IDS system while testing it using a specialized IIoT-based dataset which would simulate the real case scenario. Though, the baseline model has applied various ML techniques on the dataset. Hence, this study applied the feature selection to further enhance the accuracy performance of the ML-based models while minimizing the computational cost.

3. Materials and Methods

3.1. Proposed IIoT Intrusion Detection Method. The proposed system aims at enhancing the performance of NIDSs for IIoT-based networks using feature selection techniques on the dataset. In recent years' various techniques like data mining and ML techniques have been used to resolve various problems involving optimization system performance. To improve the performance of NIDS for IIoT-based networks, the proposed model reduces the number of features used for the classification problem. Figure 1 presents the architecture of the model that has been proposed. The stages of the

Table 1: Summary of the main findings of the related work.

Method name	Accuracy (%)	Precision	Recall	*F*-score	Feature selection	Dataset
GWO-PSO-RF NIDS model [20]	99.88% 99.24% 99.87% 99.66% (avg)	0.67 0.93 0.94	0.72 0.97 0.87	0.69 0.95 0.89	Hybrid GWO-PSO	KDDCup99 NSL-KDD CICIDS2017
DL-rule based feature selection model [3]	99.0% 98.9%	0.97 0.99	0.99 0.99	0.98 0.97	Rule based selection	NSL-KDD UNSW-NB15
GA-RF model [19]	87.61%	0.98	0.81	0.89	Genetic algorithm	UNSW-NB15
AQU-CNN [33]	99.99% 77.38% 99.99% 99.92%	0.99 0.84 0.99 0.94	0.99 0.77 0.99 0.92	0.99 0.77 0.99 0.93	Aquila optimizer (AQU).	CIC2017 NSL-KDD BoT-IoT KDD99
Attack detection using ensemble classifier [30]	83% 87% 87%	0.86 0.88 0.88	0.84 0.88 0.88	0.83 0.87 0.87	—	WUSTL_IIOT-2018 N_BaIoT Bot_IoT (Naïve Bayes)
DL-TL-NIDS [31]	99.97% 99.86% 99.97%	0.995 0.997 0.997	0.95 0.998 0.997	0.996 0.998 0.997	—	TON IoT CICIDS-2017 CICIDS-2018

proposed model were discussed in detail in the following subsection. The method consists of preprocessing, feature selection, and classification.

3.2. The Preprocessing Stage. To provide appropriate data for the proposed model framework for the model optimization, various preprocessing steps were performing on the WUSTL-IIOT-2021 dataset. The following are the steps followed to reform the dataset used for the purpose of this study:

(i) *Removing Features*. Features that are unique to the attacks are removed after downloading the dataset ('StartTime', 'LastTime','SrcAddr', 'DstAddr', 'sIpId', 'dIpId'), therefore, if not removed, the model would not be universal for unseen data since they expose the type of the attack to the model. Also, the attack cannot be included as a feature, hence, it is very necessary to remove them, and the main objective is to reduce the features of the dataset before classification

(ii) *Label Encoding*. The traffic label is given string value to specify the type of attack in which it belongs, hence, it is very necessary to change the value encoded into numerical values

(iii) *Data Binarization*. The data collected in the collection spans a wide range of values. This data presents the classifier with a variety of obstacles during the training process in order to correct such differences. As a result, each feature's values must be standardized. As a result, the lowest value for each characteristic should be 0. The maximum value, however, should be 1. It improves the homogeneity of the classifier. It keeps the discrepancy amongst each feature's values

(iv) *Addressing Imbalance Data*. This was handled using resampling without replacement with a 20% sample size model for the dataset before classification

3.3. Feature Selection. The importance of feature selection in improving the performance of NIDSs cannot be overstated because it also improves the performance of IDSs. This is due to the fact that intrusion detection involves a huge number of features that take a long time to process. Hence, feature selection is very important to increase the detection rate (DR) and decrease the detection time and false alarm rate. This problem can be solved using bioinspired optimization methods. As a result, the feature selection method influences the amount of time required to examine traffic behavior and enhance the overall performance of the model. It is very challenging to select the subset of features in any given dataset, and when the dimensionality of the feature is high, it cannot be managed efficiently. They can provide high-quality solutions in a fair amount of time and with considerable diligence [34]. Two bioinspired metaheuristic algorithms were used for the purpose of feature selection, namely, PSO [15] and BA methods [16].

3.3.1. Particle Swarm Optimization. One of the most stunning tourist attractions is a flock of birds in flight. Herds and other forms of organizations, such as plants and terrestrial animals, are fascinating to observe and consider organizational behavior. It includes a variety of birds, but the overall exercise is fluid. It is straightforward, but visually complex. It appears to be arranged at random. It is breathtaking. The feeling of deliberate and concentrated dominance is the most humiliating. Furthermore, all the data suggest that the flock's movement is solely the result of each bird's recognition of the area. Bird-like objects called boids are employed in the flocking model [35]. Each boid is known for what happens in its

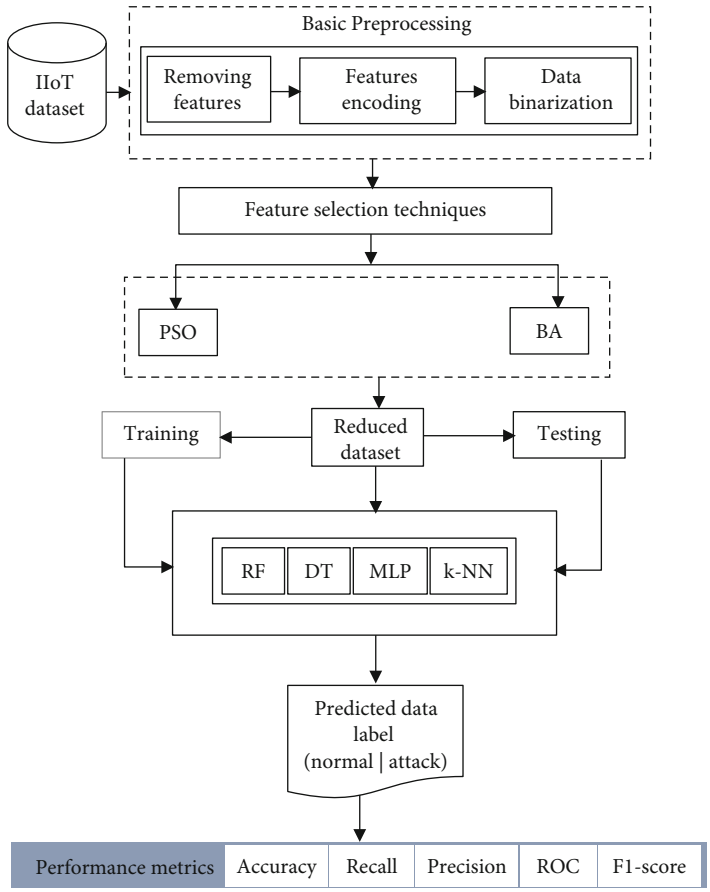


Figure 1: The proposed NIDS Architecture for IIoT.

immediate environs because of its position and speed. The three basic steering behaviors shown by boids are separation, alignment, and cohesion [36].

A PSO does not necessitate a thorough understanding of the situation, such as gradual changes [37]. It can be utilized when a problem requires access to data that is either unavailable or prohibitively expensive. Each particle's fitness score is determined in a swarm. The particle's best position is determined using a fitness score. Each particle's position indicates a potential solution to the optimization issue [38]. After that, the best global location among the particles is determined. It uses the best global and local locations to locate intriguing places for further research, as well as spots where all this information is shared with other particles, allowing particles to explore the solution space more effectively. It is a method of iterative optimization [39].

Each particle is defined by the original PSO formulas as a potential solution to a problem in $N-dimensional$ space. Particle i's position is denoted as $X_i = (x_{i1}, x_{i2}, \cdots, x_{\mathrm{iN}})$. Each component also remembers its prior optimal position, which is expressed as $P_i = (p_{i1}, p_{i2}, \cdots, p_{\mathrm{iN}})$. Because each particle in a swarm is rotating, it has a momentum, which may be expressed as $V_i = (v_{i1}, v_{i2}, \cdots, v_{\mathrm{iN}})$.

Among $p$best, each particle knows its best value so far ($p$best) and the best value in the group ($gbest$). This information is useful in determining how the particles in their immediate vicinity have done. Using the following information, each particle tries to change its position:

(i) the gap between $p$best where you are now and where you want to be

(ii) the distance between where $g$best is now and where $g$best want to be

The notion of velocity can be used to illustrate this change. Each agent's velocity can be altered (3). Eberhart and Shi were the first to mention the incorporation of an inertia weight in the PSO algorithm in the literature [40]. Consider

$$V_{id} = w \,\mathrm{x}\, V_{id} + c_i \,\mathrm{x}\, \mathrm{rand}\,(\,) \,\mathrm{x}\, (P_{id} - X_{id}) + c_2 \,\mathrm{x}\, \mathrm{rand}\,(\,) \,\mathrm{x}\, \left(p_{gd} - \mathrm{X}_{id}\right), \tag{1}$$

where the index of the particle i is, $i \in \{1, \cdots, n\}$, N population size, d dimension, $d \in \{1, \cdots, N\}$, rand ( ) is uniformly distributed random variable between 0 and 1, V_{id}: velocity of particle i on dimension d, X_{id} current position of particle on dimension d, c_1 establishes the relative importance of the cognitive process, the factor of self-confidence, and the factor of motivation, c_2 defines the social component's proportionate influence, swarm confidence factor, P_{id} personal best or $p$best

of particle i, p_{gd} global best or $g$best of the group, and w inertia weight.

The following equation can be used to change the existing position in the solution space, which is the searching point:

$$V_{id} = X_{id} + V_{id}. \quad (2)$$

Because all swarm particles tend to move towards better positions, the best position (i.e., optimum solution) can finally be attained by combining the efforts of the entire population. PSO is a basic, easy-to-implement, and computationally efficient method.

3.3.2. BAT Algorithm. This was created using the key concept of frequency tuning based on microbat echolocation. The echolocation features of microbats can be idealized as the following three rules in the typical bat algorithm:

All bats utilize echolocation to gauge distance, and in some mysterious way, they also 'know' the distance between food/prey and backdrop barriers.

Bats look for prey by flying at a random velocity v_i at position x_i with a fixed frequency $f_{\min}$, changing wavelength, and loudness A_o. Depending on the closeness of their target, they may automatically modify the wavelength (or frequency) of their radiated pulses as well as the rate of pulse emission $r \epsilon [0, 1]$.

Even though loudness can change in a variety of ways, we assume that it ranges from a high (positive) A_o to a low (constant) $A_{\min}$.

The virtual bats require the following initialization parameters: the d-dimensional search space, position x_i, velocity v_i, and frequencyf_1. The following are the update rules for the new solution x_j^i and velocity v_j^i in each step t:

$$f_i = f_{\min} + (f_{\min} - f_{\max})\beta, \quad (3)$$

$$v_i^j = v_i^j(t-1) + \left[\widehat{x^j} - x_i^j(t-1)\right] f_{i,} \quad (4)$$

$$x_i^j(t) = x_i^j(t-1) + v_i^j, \quad (5)$$

where $\beta \epsilon [0, 1]$ denotes a uniformly distributed random vector. We know that the variable f_i is utilized to change the velocity and that the variable $x_i^j(t)$ represents the value of the position j for the bat i at the step t based on Equations (3), (4), and (5). The variable x^j denotes the current global best position, which is determined by comparing all of the m bats' answers.

Song and Gorla used a random walk technique for each bat to prevent them from falling into the local extremum and to boost their random searching ability [41]. Following the selection of a solution from the current best position, the random walk is used to generate a new solution for each bat, as described in

$$x_{\text{new}} = x_{\text{old}} + \varepsilon \bar{A}(t), \quad (6)$$

where $\epsilon \varepsilon [-1, 1]$ is a random number that controls the walk's direction and stride, and $\bar{A}(t)$ is the average volume of all bats in the step t.

In addition, according to Equation (7), the loudness A_i and the pulse rate r_i are updated for each step in Equation (5). When the prey is discovered, the loudness A_i is normally reduced and the pulse rate r_i is raised. For added convenience, the volume can be modified to any value.

$$A_i(t+1) = \alpha A_i(t), \quad (7)$$

$$r_i(t+1) = r_i(0)[1 - \exp(-\gamma t)], \quad (8)$$

where and are both constants The loudness $A_i(0)$ and the pulse rate $r_i(0)$ are normally chosen at random in the first phase of the bat algorithm. In general, $A_i(0) \varepsilon [1, 2]$ and $r_i(0) \varepsilon [0, 1]$ are set.

3.4. The Classifiers Models

3.4.1. Random Forest. This Bagging classifier uses a technique known as bootstrap aggregation, which is a form of ensemble technique. A number of different basic models $(M_1, M_2, \cdots, M_n)$ are blended. Using row sampling with replacement, distinct samples of records are delivered to each model. Some records may be repeated in the samples delivered to the models when row sampling with replacement is used. The voting classifier is used to combine the model outputs in order to make a judgment. A random forest is a bagging classifier in which numerous decision trees are utilized as models. Row and column sampling are used to provide input to each decision tree. The difficulty with the decision tree is that it has a low bias and a big variance. This indicates that the tree performs better in the training phase but poorly in the testing phase. The voting strategy lowers variance from high to low since the decision is based on the voting of numerous trees rather than a single tree [42].

3.4.2. Multilayer Perceptron (MLP). This is a type of ANN that feeds back information. The name MLP is confusing, referring to networks built of multiple layers of perceptrons (with threshold activation) in some cases and any feedforward ANN in others [43]. Multilayer perceptrons, especially those with a single hidden layer, are commonly referred to as "vanilla" neural networks [44]. There are at least three levels of nodes in an MLP: an input layer, a hidden layer, and an output layer. Each node, with the exception of the input nodes, is a neuron with a nonlinear activation function. Backpropagation is a supervised learning technique used by MLP during training. MLP is distinguished from a linear perceptron by its numerous layers and nonlinear activation. It can distinguish between data that is not linearly separable and data that is nonlinearly separable. If all of the neurons in a multilayer perceptron have a linear activation function, that is, a linear function that maps the weighted inputs to each neuron's output, then linear algebra shows that any number of layers may be reduced to a two-layer input-output model. In some MLP neurons, a nonlinear activation function is used that was made to model how often biological neurons fire or send out action potentials or pulses [45].

3.4.3. K-NN Algorithms. In classification and regression issues, the K-NN algorithm is used. It is a supervised learning technique that classifies an unknown instance based on the distance between the instance and k selected neighbors, with the class determined by the majority of neighbors voting [46]. The K-NN algorithm is frequently used in classification, with the goal of classifying new objects based on attributes and training examples. The K-NN technique is a classification approach based on learning data that is closest to the object. The K-NN algorithm is frequently used in classification, with the goal of classifying new objects based on attributes and training examples. The K-NN technique is a classification approach based on learning data that is closest to the object. This area is divided into divisions based on the training data's class label. A point in this space is designated as c class; if class c is the most frequently occurring point at k, then c is the correct answer. The Euclidean distance is used to determine how close or remote neighbors are [47].

4. Results and Discussion

4.1. The Dataset. The dataset used for the purpose of this study is WUSTL-IIoT-2021. This dataset consists of network data of IIoT-based systems that can be used for cybersecurity research. The dataset was captured using the IIoT testbed and presented by the authors in [48]. The goal of this testbed is to mimic real-world industrial systems as closely as possible while also allowing for real-world cyber-attacks. A total of 2.7GB of data was collected, spending about 53 hours. There are levels of preprocessing to clean the dataset by removing the rows with missing values, extreme outliers, and invalid entries resulting from corrupted values. After the preprocess stages, the final version is a little over 400 MB and can be used for the purpose of an intrusion detection experiment. Table 2 shows the statistics of the dataset.

The average data rate was 419 kbit/s, and the average packet size was 76.75 bytes, as shown in Table 3. This was purposefully focused around 90% of the attacks to DoS attacks because they are typically high in traffic and number of samples. Other forms of attacks are less common, and when they do occur, they simply convey a small amount of traffic data.

4.2. Evaluation Metrics. To assess the performance of the proposed model, the metrics in Equations employ many features, namely true positive (tp), false positive (fp), true negative (tn), and false negative (fn) [1]. The confusion matrix is a table that calculates the metric features as illustrated by Table 4 that estimates the true positive rate (TPR), false negative rate (FNR), true negative rate (TNR), and false positive rate (FPR). The main model assessors in Equations were derived from the table.

TPR is the ratio of class a instances correctly classified as class a as shown by

$$\text{TPR} = \text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}. \tag{9}$$

Table 2: The dataset characteristics.

Dataset	WUSTL-IIoT
Number of observations	1,194,464
Number of features	41
Number of attack samples	87,016
Number of normal samples	1,107,448

Table 3: Statistical information of the traffic types in our developed dataset.

Traffic's type	Percentage (%)
Normal traffic	92.72
Total attack traffic	7.28
Command injection traffic	0.31
DoS traffic	89.98
Reconnaissance traffic	9.46
Backdoor traffic	0.25

Table 4: Confusion matrix.

		Predicted Class	
		Class a	Class b
Real Class	Class a	tp	fn
	Class b	fp	tn

TNR is the ratio of class b instances correctly classified as class b as shown by

$$\text{TNR} = \frac{\text{tn}}{\text{tn} + \text{fp}}. \tag{10}$$

FPR is the ratio of class a instances incorrectly classified as class b as shown by

$$\text{FPR} = \frac{\text{fp}}{\text{fp} + \text{tn}}. \tag{11}$$

FNR is the ratio of class b instances incorrectly classified as class a as shown by

$$\text{FNR} = \frac{\text{tp}}{\text{tp} + \text{fn}}. \tag{12}$$

Accuracy is the percentage of correctly classified instances as presented by

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}. \tag{13}$$

Precision is the ratio of the number of correct decisions made as shown by

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}. \tag{14}$$

Sensitivity is ratio of the number of tp by the number of all of the positive evaluations as shown by Equations ((15a) and (15b))

$$\text{Sensitivity} = \frac{\text{tp}}{\text{tp} + \text{fn}}. \tag{15a}$$

$$\text{Specificity} = \frac{\text{tn}}{\text{tn} + \text{fp}}. \tag{15b}$$

The F1-Score is the harmonic mean between the recall and precision as illustrated by

$$F1_{\text{Score}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{16}$$

Geometric Mean is the square root of the product of sensitivity and specificity as shown by

$$\text{Geo Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{17}$$

ROC shows the tradeoff between TPR and FPR as shown by

$$\text{ROC} = \frac{1 + (\text{TPR} - \text{FPR})}{2}. \tag{18}$$

PRC shows the trade-off between precision and recall for different threshold. A high PRC value shows both high recall and precision. It is a useful assessor especially when the classes are imbalanced.

Logistic Loss (Log Loss) measure the classification model performance based on the predicted probabilities of the real class. This value increases as the probability diverges from the real label. So, the lower the value, the better the performance of the model. The formula for Log Loss for multiclass classification is shown by

$$\text{LogLoss} = -\sum_{c=1}^{M} y_{o,c} \log\left(p_{o,c}\right) \tag{19}$$

Where M is the number of labels, log is the natural log, y is the class label, p is the predicted probability observation of o is of class c.

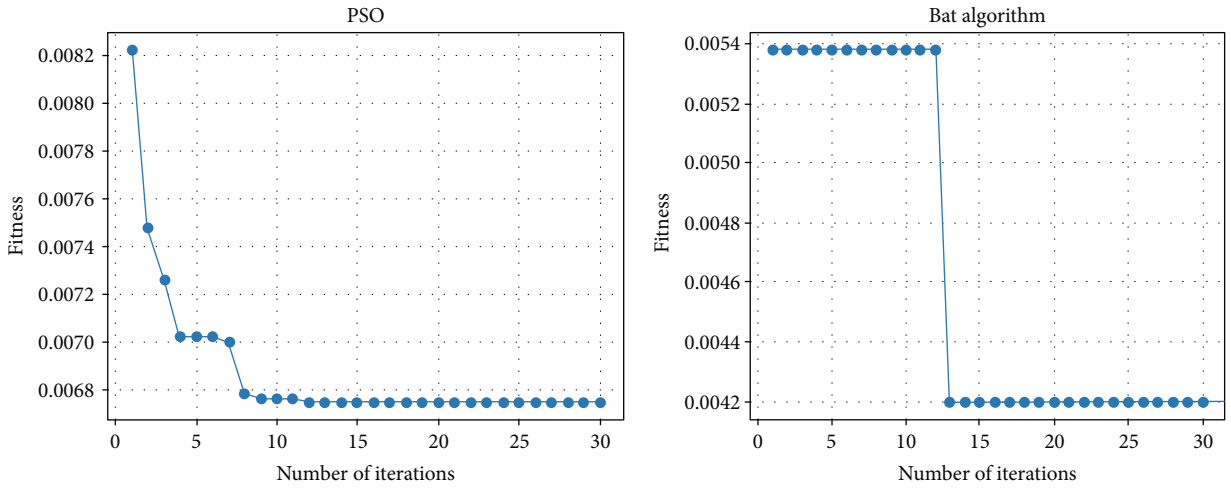
4.3. Feature Selection Schemes Results. All experiments were performed on a i7-8750H CPU @ 2.20GHz, 32 GB RAM and Windows 11 Pro. system. The study dataset was split into 80-20 of train-test ration. The general parameters used by all feature selection scheme (FSS) are k (4) which is the k -value in K-NN, the number of particles ($N = 10$) and the maximum number of iterations ($T = 30$). After application of the feature selection scheme, DstPkts, SrcBytes, DstLoss, pLoss, TcpRtt, IdleTime, and TotAppByte features were common to PSO and BA. Figures 2(a) and 2(b) show rate of convergence of the fitness function of PSO and BA. Table 5 presents the optimum features that were selected for efficient classification performance by feature selection schemes considered in this study for detecting the attacks.

Figures 2(a) and 2(b) shows the rate of convergence of the fitness function of PSO and BA for the study dataset. For PSO, the convergence happens at the 8th iteration and the best fitness value is 0.00458. PSO started with the highest fitness value of 0.0082 and at the 2nd iteration, the PSO scheme did level of exploration and gradually switched between exploration and exploitation which converges at the 8th iteration. Likewise, for BA scheme, the convergence occurs at the 13th iteration with the best fitness value is 0.00419. The highest fitness of 0.00537 is steeply decreased by switching between exploration and exploitation. So, at their individual best fitness function, the schemes make the search for the global optimal solution.

4.4. Evaluation Results of the Classifiers. The proposed model is assessed based on the RF, K-NN, and MLP machine learning classifiers. The results of the performed experiment that is based on the aforementioned classifiers are presented in Table 6. These outcomes are based on the two-feature selection scheme (PSO and BA) adopted for the study and RF, K-NN, and MLP. It is observed that the highest recall rate of 0.996 was obtained from RF based on the dataset created from BA scheme which was closely followed by RF on the original dataset with a value of 0.98. Likewise, for accuracy metric as observed from Table 5, it is observed that RF on dataset created from BA scheme scored the highest value of 99.99%. Similarly, for F1_Score and Precision metrics, RF on dataset created from BA scheme still scored the same highest values of 0.996 and 0.996, respectively. So, based on classification report, RF on dataset created from BA scheme gave a superior performance compared to other models. MLP classifier performed poorly for both schemes and the aforementioned metrics.

Table 7 presented the result of the dataset analysis metrics efficient for evaluation of imbalanced dataset. Since sampling without replacing to the tune of 20% was applied to the dataset to treat the imbalance, geometric mean, Precision-Recall-curve, and log loss are better suited metrics. Based on geometric mean, RF gave a superior performance based on BA scheme with a value of 0.996 while RF and K-NN scored the value of 1 on dataset created from BA scheme for PRC, Log Loss and ROC metric, respectively. MLP classifier performed poorly for both schemes and the aforementioned metrics. These metrics was used since they are best in measuring the imbalanced cases. The results revealed that BA with RF performed better across all the performance metrics when compare with the PSO feature selection algorithm. The best of all the classifiers is the RF with the BA classifier.

4.4.1. Confusion Matrix (CM). The table of CM is used to define the performance of any classification models. This is used here to visualize and summarize the results of the performance of the proposed classifiers. This CM table shows the detection rate of each of the classes. Based on the results presented in Tables 5 and 6, it could be deduced that RF and DTC produced results that were similar. So, further analysis to reveal which model and on which performed best. Figures 3(a)–3(c) are confusion



(a) Fitness function for PSO

(b) Fitness function for BA

Figure 2: The convergence of the fitness function of PSO and BA.

Table 5: Selected Features.

Name of FSS	Parameters	Fitness value	No. of features	Index of selected features
Particle swarm optimization (PSO)	w = 0.9 c1 = 0.5 c2 = 0.5	0.00458	17	[mean, DstPkts, SrcBytes, TotBytes, SrcLoad, DstLoad, SrcRate, SrcLoss, DstLoss, pLoss, proto, TcpRtt, IdleTime, max, sTtl, TotAppByte, SynAck]

Bat algorithm (BA)	Fmax = 2 # maximum frequency
	Fmin = 0 # minimum frequency
	Alpha = 0.9 # constant
	Gamma = 0.9 # constant
	A = 2 # maximum loudness
	r = 1 # maximum pulse rate
		0.00419	16	[sport, Dport, SrcPkts, Dst, Pkts, SrcBytes, DstLoss, loss, pLoss, DstJitter, Tcp, Rtt, IdleTime, sum, min, sDSb, TotAppByte, DstJitAct]

Table 6: Classification report.

Assessor	FSS	RF	K-NN	MLP
Recall	PSO	0.958	0.94	0.792
	BA	0.996	0.99	0.819
	Original	0.98	0.94	0.34
Precision	PSO	0.954	0.946	0.882
	BA	0.996	0.966	0.94
	Original	0.992	0.974	0.579
Accuracy	PSO	0.997	0.995	0.967
	BA	0.999	0.998	0.459
	Original	0.993	0.975	0.467
F1_Score	PSO	0.956	0.942	0.832
	BA	0.996	0.978	0.266
	Original	0.982	0.951	0.297

Table 7: Imbalance report.

Assessor	FSS	RF	K-NN	MLP
Geo mean	PSO	0.978	0.97	0.878
	BA	0.996	0.996	0.162
	Original	0.992	0.968	0.254
Log loss	PSO	0.036	0.037	1.024
	BA	0.0075	0.0075	18.66
	Original	0.0008	0.042	18.3309
PRC	PSO	0.0081	0.072	18.3021
	BA	1	1	0.319
	Original	1	0.998	0.326
ROC	PSO	0.98	0.98	0.88
	BA	1	1	0.55
	Original	1	0.99	0.57

matrixes based on the original and the dataset obtained from feature selected schemes (PSO, and BA) for and RF, K-NN, and MLP models. The class labels: Backdoor, CommInj, DoS, Reconn, and normal represented by 0.0, 1.0, 2.0, 3.0, and 4.0. From Figures 3(a) which presents BA scheme on RF, the classification performance for class Backdoor represented by 0.0 was 100% because all 54 instances were correctly classified. For class label CommInj represented by 1.0, 66 instances were correctly

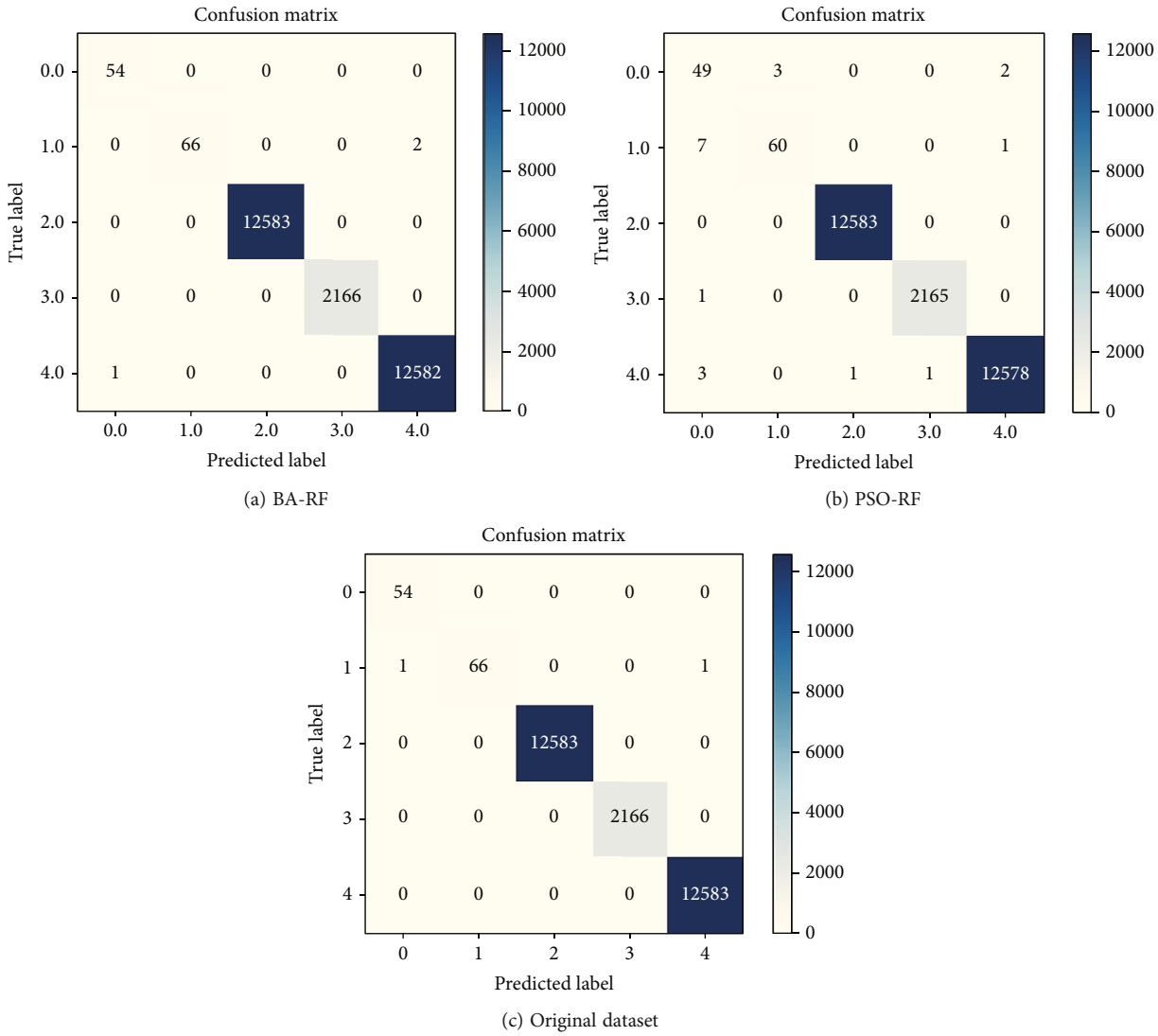


Figure 3: (a), (b), and (c) showing confusion matrix for BA, PSO, and original dataset on RF.

classified out of 68, 2 instances were misclassified as normal. For DoS:2.0 and Reconn:3.0, their classification performance was 100%. For normal:4.0 class which is the majority class, 12582 instances were rightly classified out of 12583 instances while 1 instance was misclassified as Backdoor attack.

Similarly, for Figure 3(b) which presents PSO scheme on RF, out of 54 instances for class Backdoor represented by 0.0, only 49 instances were correctly classified, 3 instances were misclassified as CommInj and 2 instances as normal. For class label CommInj represented by 1.0, 60 instances were correctly classified out of 68, 7 instances were misclassified as Backdoor while 1 instance is misclassified as normal. For DoS:2.0, the classification was 100%. For Reconn:3.0, 2165 out of 2166 were rightly classified while 1 was misclassified as Backdoor. For normal:4.0 class which is the majority class, 12578 instances were rightly classified out of 12583 instances while 3 instances were misclassified as Backdoor, 1 instance were wrongly classified as DoS and Reconn, respectively.

From Figure 3(c) which presents original dataset on RF model, the classification performance for class Backdoor represented by 0.0 was 100% because all 54 instances were correctly classified. For class label CommInj represented by 1.0, 66 instances were correctly classified out of 68, 1 instance were misclassified as Backdoor and 1 instance were misclassified as normal. For DoS:2.0, Reconn:3.0, and normal:4.0, their classification performance was 100%.

4.4.2. Comparison of the Proposed Model with Existing Models. In recent years, researchers have attempted to resolve the issues of intrusion detection in the IoT network. As mentioned earlier, these researches are carried out using various techniques such as ML, semisupervised learning, adaptive, heuristic, decision tree, and DL. However, bioinspired algorithms have been employed to extract the relevant features of the IoT networks in order to decrease processing cost, memory, and pave a smooth way to apply various classification techniques from the selected features. In this section, we present the related works on handling intrusion detection in the IIoT that employs bioinspired algorithms. At last, they are summarized in Table 8 comparatively.

Table 8: Comparison of the proposed model with state-of-the-art with existing models.

Method name	Accuracy (%)	Precision	Recall	*F*-score	Feature selection	Dataset
GWO-PSO-RF NIDS model [20]	99.88%	0.67	0.72	0.69	Hybrid GWO-PSO	KDDCup99
	99.24%	0.93	0.97	0.95		NSL-KDD
	99.87%	0.94	0.87	0.89		CICIDS2017
	99.66% (avg)					
DL-Rule Based Feature Selection Model [3]	99.0%	0.97	0.99	0.98	Rule based selection	NSL-KDD
	98.9%	0.99	0.99	0.97		UNSW-NB15
GA-RF model [19]	87.61%	0.98	0.81	0.89	Genetic algorithm	UNSW-NB15
AQU-CNN [33]	99.99%	0.99	0.99	0.99	Aquila optimizer (AQU).	CIC2017
	77.38%	0.84	0.77	0.77		NSL-KDD
	99.99%	0.99	0.99	0.99		BoT-IoT
	99.92%	0.94	0.92	0.93		KDD99
Proposed model	99.99%	0.996	0.996	0.996	BA	WUSTL-IIoT
	95.68%	0.997	0.958	0.956	PSO	

In [20], the authors used an hybrid GWO-PSO for feature selection before employing RF for classification of the dataset used to test the performance of the proposed model. The model performs reasonable better with average of 99.66%, but the proposed model still performs better with BA used for feature selection on the dataset used. The authors in [19] had used the genetic algorithm along with random forest model which was employed in the fitness function of the genetic algorithm proposed an IDS for IIOT. The reason behind the use of GA is the presence of the high number of features in the modern datasets as well as the number of network traces. As a result, the training process of the ML algorithms are negatively impacted and mislead as the ML performance reduces as the feature numbers increases. It is harder to perform the learning process as the number of attributes increases in the dataset. Therefore, the genetic algorithm is used for enhancing the feature selection and for each attribute vector, the author implemented Tree-based algorithms such as RF, DT, and ET algorithms which is conducted on the UNSW-NB15 general-purpose dataset.

The same dataset has been used along with the NSL-KDD dataset to implement a hybrid rule-based feature selection approach by authors in [3]. The proposed study integrates deep feedforward neural network model and rule-based feature selection with the applications of the IIOT in order to gather the relevant information that can be used to develop an intelligent NIDS (i.e., information is captured from TCP/IP packets). This study is a three-tier model for intrusion detection in IIoT systems in which a rule-based model is used for feature selection along with a genetic tool were used for feature selection and to generate attributes with the greatest values. At the end, the features that have been selected are loaded into the ANN for learning purposes.

The authors in [33] presented using feature selection for IDS in IoT-based system to remove irrelevant parameters before applying the DL model on the dataset. The proposed performed very well on the dataset used with an accuracy of 99.99%, and the model reduced the computational time of the proposed system. The proposed model performance was reasonably well with compared to the existing similar work in IoT-based systems. The accuracy of BA feature selection did well when compared with the existing models in this area.

Table 9: Comparison of the proposed model with baseline model.

Technique	Accuracy (%)	Precision (%)	Feature selection
[49]	99.99	99.95	—
[2]	98.40	—	—
[48]	99.99	97.44	—
Proposed model	99.99	99.96	BA

To really show the importance of employing feature selection on the dataset before classification models, the proposed model used the baseline methods to compare the proposed model. Table 9 displays the comparison of the proposed model with the baseline model that used and created the dataset used.

In [49], RF performs better when compared with other ML-models used for the classification of the dataset with 99.99%, and Naïve Bayes has the least performance in term of accuracy with 97.48%, both RF and Naïve Bayes perform better in term of precision with 97.44%, but according to the authors, accuracy is not the best performance metric when it comes to the classification of huge amount of data, the sensitivity (precision) metric. Therefore, it can be said that the proposed model using feature selection with RF performs better than the baseline models. The computational time of the proposed models is very fast since the number of parameters used is reasonably reduced when compare with the baseline model. The same authors in [2] recorded an accuracy of 99.99%, and 99.95% of precision. In another study by the same authors in [48], the RF and Naïve Bayes performs better with precision of 97.44%, and the least of all the classifiers is the Logistic Regression with 47.44%.

Therefore, the proposed model performs reasonably better in terms of precision when compared with the baseline model. Hence, the model is optimal when in use in a real-world IIoT-based environment with huge amounts of unstructured and unlabeled datasets. The use of feature selection greatly reduces the computational time used in processing the dataset when

compared with the baseline, thus automatically reduces the data dimensionality and examines high-level functionality with effective accuracy and precision. Although our results seem similar to the other related work, as can be seen in Table 8, our proposed method has been tested on a more relevant dataset, WUSTL-IIoT, which is specifically collected for IIoT environment. So, our results would be more reliable than other related work.

5. Conclusions

The emergence of various cybersecurity techniques associated with IIoT-based network traffic has become critical to securing the IIoT environment from attackers and intruders from the outside world. Big data enabled with ML-based classifiers is a powerful tool for the analysis of huge data with the intention of securing the IIoT technology. The technologies have been proven helpful in the security of the IIoT-based system. However, the divergent implications and fundamental differences between IACS and traditional IT systems for counter-cyber-attacks are distinct. Thus, special attention is required to provide security for the IIoT. Therefore, this study proposes a feature selection scheme with ML-based models for the classification of NIDS in IIoT-based traffic. The PSO and BA are used for feature selection to reduce the parameters used for the classification of the IIoT-based dataset used. For the classification, three different ML-based models are used to classify the dataset. The ML techniques were used to handle the new types of attacks like command injection, SQL injection, and backdoors after applying the feature section schemes to the dataset. The dataset used for the proposed model is the WUSTL-IIoT cybersecurity research. The experimental results show that the proposed model performs greatly better when compared with the baseline model, which created the testbed dataset with an accuracy of 99.99%, and 99.96% for precision. The feature extraction on the dataset reduces the computational time of the proposed model, which is very necessary when considering the use of an IIoT-based system. Future work will consider the use of a deep learning model for the classification of the dataset for ranking the attack traffic from the normal traffic. The security of the proposed system can be enhanced using the blockchain and various encryption techniques.

Data Availability

The data used in the study can be found in: https://www.cse.wustl.edu/~jain/iiot2/index.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study is supported via funding from the Prince Sattam bin Abdulaziz University (project number PSAU/2023/R/1444).

References

[1] M. Iaiani, A. Tugnoli, S. Bonvicini, and V. Cozzani, "Analysis of cybersecurity-related incidents in the process industry," *Reliability Engineering & System Safety*, vol. 209, article 107485, 2021.

[2] M. Zolanvari, M. A. Teixeira, and R. Jain, "Effect of imbalanced datasets on security of industrial IoT using machine learning," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 112–117, Miami, FL, USA, November 2018.

[3] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7154587, 17 pages, 2021.

[4] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.

[5] S. Gibbs, "Triton: hackers take out safety systems in'watershed'attack on energy plant," *The Guardian*, 2017, https://www.theguardian.com/technology/2017/dec/15/triton-hackers-malware-attack-safety-systems-energyplant.

[6] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST Special Publication*, vol. 800, no. 82, pp. 16–16, 2011.

[7] S. Varghese, *Digital Twin-based Intrusion Detection for Industrial Control Systems*, 2021.

[8] T. Gaber, A. El-Ghamry, and A. E. Hassanien, "Injection attack detection using machine learning for smart IoT applications," *Physical Communication*, vol. 52, article 101685, 2022.

[9] R. O. Ogundokun, J. B. Awotunde, P. Sadiku, E. A. Adeniyi, M. Abiodun, and O. I. Dauda, "An enhanced intrusion detection system using particle swarm optimization feature extraction technique," *Procedia Computer Science*, vol. 193, pp. 504–512, 2021.

[10] J. B. Awotunde and S. Misra, "Feature extraction and artificial intelligence-based intrusion detection model for a secure internet of things networks," in *Illumination of Artificial Intelligence in Cybersecurity and Forensics*, pp. 21–44, Springer, 2022.

[11] M. Al-Omari, M. Rawashdeh, F. Qutaishat, M. Alshira'H, and N. Ababneh, "An intelligent tree-based intrusion detection model for cyber security," *Journal of Network and Systems Management*, vol. 29, no. 2, pp. 1–8, 2021.

[12] J. B. Awotunde, R. G. Jimoh, S. O. Folorunso, E. A. Adeniyi, K. M. Abiodun, and O. O. Banjo, "Privacy and security concerns in IoT-based healthcare systems," in *The Fusion of Internet of Things, Artificial Intelligence, and Cloud Computing in Health Care*, pp. 105–134, Springer, Cham, 2021.

[13] S. S. Mohammed, R. Hussain, O. Senko et al., "A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–8, Limassol, Cyprus, October 2018.

[14] J. B. Awotunde, K. M. Abiodun, E. A. Adeniyi, S. O. Folorunso, and R. G. Jimoh, "A deep learning-based intrusion detection technique for a secured IoMT system," in *International Conference on Informatics and Intelligent Applications*, pp. 50–62, Springer, 2021.

[15] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[16] X. S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[17] A. S. Lalos, A. P. Kalogeras, C. Koulamas, C. Tselios, C. Alexakos, and D. Serpanos, "Secure and safe IIoT systems via machine and deep learning approaches," in *Security and Quality in Cyber-Physical Systems Engineering*, S. Biffl, M. Eckhart, A. Lüder, and E. Weippl, Eds., pp. 443–470, Springer, Cham, 2019.

[18] B. Valeske, A. Osman, F. Römer, and R. Tschuncky, "Next generation NDE sensor systems as IIoT elements of industry 4.0," *Research in Nondestructive Evaluation*, vol. 31, no. 5-6, pp. 340–369, 2020.

[19] S. M. Kasongo, "An advanced intrusion detection system for IIoT based on GA and tree based algorithms," *IEEE Access*, vol. 9, pp. 113199–113212, 2021.

[20] P. K. Keserwani, M. C. Govil, E. S. Pilli, and P. Govil, "A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model," *Journal of Reliable Intelligent Environments*, vol. 7, no. 1, pp. 3–21, 2021.

[21] F. E. Ayo, S. O. Folorunso, A. A. Abayomi-Alli, A. O. Adekunle, and J. B. Awotunde, "Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection," *Information Security Journal: A Global Perspective*, vol. 29, no. 6, pp. 267–283, 2020.

[22] D. Moon, H. Im, I. Kim, and J. H. Park, "DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 2881–2895, 2017.

[23] G. Prethija and J. Katiravan, "Machine learning and deep learning approaches for intrusion detection: a comparative study," in *Inventive Communication and Computational Technologies*, pp. 75–95, Springer, 2022.

[24] O. A. Abisoye, O. S. Akanji, B. O. Abisoye, and J. Awotunde, "Slow hypertext transfer protocol mitigation model in software defined networks," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, pp. 1–5, Sakheer, Bahrain, October 2020.

[25] M. Mayuranathan, M. Murugan, and V. Dhanakoti, "Retracted article: best features based intrusion detection system by RBM model for detecting DDoS in cloud environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3609–3619, 2021.

[26] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Generation Computer Systems*, vol. 113, pp. 418–427, 2020.

[27] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *Ict Express*, vol. 4, no. 2, pp. 95–99, 2018.

[28] W. Lian, G. Nie, B. Jia, D. Shi, Q. Fan, and Y. Liang, "An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning," *Mathematical Problems in Engineering*, vol. 2020, Article ID 2835023, 15 pages, 2020.

[29] C. Long, Y. Zhang, J. Wei, W. Wan, J. Zhao, and G. Du, "A hybrid intrusion detection algorithm based on Gaussian mixture model and nearest neighbors," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, pp. 117–120, Osnabrueck, Germany, October 2019.

[30] V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif, and E. A. Nasr, "Robust attack detection approach for IIoT using ensemble classifier," https://arxiv.org/abs/2102.01515.

[31] K. Raja, K. Karthikeyan, B. Abilash, K. Dev, and G. Raja, "Deep learning based attack detection in IIoT using two-level intrusion detection system," 2021.

[32] P. K. Keserwani, M. C. Govil, and S. E. Pilli, "An optimal intrusion detection system using GWO-CSA-DSAE model," *Cyber-Physical Systems*, vol. 7, no. 4, pp. 197–220, 2021.

[33] A. Fatani, A. Dahou, M. A. A. al-qaness, S. Lu, and M. A. Abd Elaziz, "Advanced feature extraction and selection approach using deep learning and Aquila optimizer for IoT intrusion detection system," *Sensors*, vol. 22, no. 1, p. 140, 2021.

[34] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, p. 1046, 2020.

[35] A. P. Bhopale and A. Tiwari, "Swarm optimized cluster based framework for information retrieval," *Expert Systems with Applications*, vol. 154, article 113441, 2020.

[36] M. M. Khan, K. Kasmarik, and M. Barlow, "Autonomous detection of collective behaviours in swarms," *Swarm and Evolutionary Computation*, vol. 57, article 100715, 2020.

[37] A. Tharwat, T. Gaber, A. E. Hassanien, and B. E. Elnaghi, "Particle swarm optimization: a tutorial," in *Handbook of Research on Machine Learning Innovations and Trends*, A. Hassanien and T. Gaber, Eds., pp. 614–635, IGI Global, 2017.

[38] W. Niu, Z. Feng, C. Cheng, and X. Wu, "A parallel multi-objective particle swarm optimization for cascade hydropower reservoir operation in Southwest China," *Applied Soft Computing*, vol. 70, pp. 562–575, 2018.

[39] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimization: theory, analysis, and application," *Swarm and Evolutionary Computation*, vol. 63, article 100868, 2021.

[40] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, pp. 84–88, La Jolla, CA, USA, July 2000.

[41] F. Liu, X. Yan, and Y. Lu, "Feature selection for image steganalysis using binary bat algorithm," *IEEE Access*, vol. 8, pp. 4244–4249, 2019.

[42] S. Y. Kim and A. Upneja, "Majority voting ensemble with a decision trees for business failure prediction during economic downturns," *Journal of Innovation & Knowledge*, vol. 6, no. 2, pp. 112–123, 2021.

[43] N. V. Sailaja, M. Yelamarthi, Y. H. Chandana, P. Karadi, and S. Yedla, "Early detection of sepsis on clinical data using multi-layer perceptron," in *Machine Learning Technologies and Applications*, pp. 223–233, Springer, 2021.

[44] S. I. Altelbany and A. A. Abualhussein, "Performance comparison of neural networks (MLP, RBFNN, ERNN, JRNN) models for stock prices forecasting to Bank of Palestine," *Muthanna Journal of Administrative and Economic Sciences*, vol. 11, no. 1, pp. 8–28, 2021.

[45] N. Wagh and S. D. Agashe, "Data-driven frameworks for system identification of a steam generator," in *Data Management, Analytics and Innovation*, pp. 441–452, Springer, 2022.

[46] M. A. Fauzi, A. T. Hanuranto, and C. Setianingsih, "Intrusion detection system using genetic algorithm and K-NN algorithm on DoS attack," in *2020 2nd International Conference on*

Cybernetics and Intelligent System (ICORIS), pp. 1–6, Manado, Indonesia, October 2020.

[47] J. K. Seth and S. Chandra, "MIDS: Metaheuristic based intrusion detection system for cloud using k-NN and MGWO," in *International Conference on Advances in Computing and Data Sciences*, pp. 411–420, Springer, 2018.

[48] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6822–6834, 2019.

[49] M. Zolanvari, A. Ghubaish, and R. Jain, "ADDAI: anomaly detection using distributed AI," in *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1–6, Xiamen, China, December 2021.