Decision Analytics Journal 7 (2023) 100206

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A metaheuristic-based ensemble feature selection framework for cyber threat detection in IoT-enabled networks

Arun Kumar Dey, Govind P. Gupta *, Satya Prakash Sahu

Department of Information Technology, National Institute of Technology, Raipur, C.G., India

ARTICLE INFO

Keywords:
Metaheuristic
Cybersecurity
Internet of Things
Feature selection
Ensemble learning
Cyber threat detection

ABSTRACT

Internet of Things (IoT) enabled networks are highly vulnerable to cyber threats due to insecure wireless communication, resource constraint architecture, different types of IoT devices, and a high volume of sensor data being transported across the network. Therefore, IoT-compatible cybersecurity solutions are required. An intrusion detection system is one of the most common solutions for detecting cyber threats in IoT-enabled networks. However, most of the existing solutions for cyber threat detection suffer from many issues like poor accuracy, high learning complexity, low scalability, and high false positive rate (FPR). We propose a metaheuristic-based intelligent and novel framework for cyber threat detection using ensemble feature selection and classification approaches to overcome these issues. First, a metaheuristic-based ensemble feature selection framework is designed using Binary Gravitational Search Algorithm (BGSA) and Binary Grey Wolf Optimization (BGWO) to get an optimized set of features to avoid the curse of dimensionality for efficient learning. Next, Decision Tree and ensemble learning-based classification techniques such as AdaBoost and Random Forest (RF) are employed separately to detect and classify cyber threats. The UNSW-NB15 dataset assesses the effectiveness of the proposed framework, and its performance is evaluated against recent state-of-the-art frameworks. Based on the result analysis, it is found that the RF outperforms existing modern cyber threats detection methods due to the optimized feature subset (4 features out of 42), maximum accuracy (99.41%), maximum detection rate (99.09%), and maximum F1-score (99.33%) with the lowest FPR (0.03%).

1. Introduction

In IoT-enabled networks, the different types of IoT devices are connected via a distributed network that enables developers to design and develop smart environments and smart systems. IoT systems permit intelligent gadgets embedded in common things to communicate and exchange sensitive information and participate in performing cooperative tasks [1]. IoT-enabled networks have been used in various areas such as the Internet of Medical Things, the Internet of Industrial Things, IoT-enabled Intelligent Transportation, and IoT-enabled Intelligent Agriculture, etc. The application of IoT helps society in many aspects, such as intelligent homes, intelligent cities, intelligent transportation, intelligent health care, and many more [2,3]. The purpose of such an intelligent environment is to increase the value and efficiency of human life by resolving difficulties associated with their living conditions. The IoT-based intelligent environment has also led to various security threats. The term cyber threats refer to any harmful activity that wants to steal data or hurt resources [4]. Moreover, cyber threats have caused economic losses and weakened IoT-based infrastructures. Thus, cyber threats detection is a fundamental research issue for the IoT-enabled network.

Recent years have witnessed various cyber-attacks such as in 2016, a cyber-attack known as Dyn exploited IoT devices embedded within intelligent homes to operate as botnets using malware named Mirai [5]. Similarly, in the year 2017, a cyber threat named Silex Malware infected 1650 IoT devices [6]. Consequently, a manual reinstallation of the firmware was performed by the owner to restore the device's default settings. In the year 2020, three gas pipeline companies in the U.S. announced another cyber-attack, claiming that electronic communication systems had been down for several days [7]. In June 2022, an HTTPS Distributed Denial-of-Service cyber-attack reached its peak when a user of "Google Cloud Armor" was attacked with 46 million requests/second which was recorded as the world's largest cyber threat [8]. However, this attack's traffic was analyzed and detected by Google early in its lifecycle. In addition, Statista reports that by 2025, the number of connected IoT gadgets will rise to 75.44 billion [9]. It will lead to a gain of 11 trillion USD annually in the economy by 2025 (approximately 11% of the global economy) [10]. Therefore, a robust and more secure intelligent cyber threat detection framework is required to protect IoT-enabled networks from both known as well as contemporary threats.

* Corresponding author.
E-mail addresses: akdey.phd2020.it@nitrr.ac.in (A.K. Dey), gpgupta.it@nitrr.ac.in (G.P. Gupta), spsahu.it@nitrr.ac.in (S.P. Sahu).

https://doi.org/10.1016/j.dajour.2023.100206
Received 15 January 2023; Received in revised form 3 March 2023; Accepted 21 March 2023
Available online 24 March 2023

Amidst technological advances, the above-discussed attacks pose a serious threat to computer security guidelines such as Confidentiality, Integrity, and Availability [11]. Moreover, despite using existing hardware and software-based security solutions such as anti-malware, firewalls, IDS, and intrusion prevention systems, many organizations are still vulnerable to cyber threats [12]. This is due to a lack of robustness in these solutions when used to protect against contemporary cyber threats as well as known threats [13]. Thus, it is crucial to design an intelligent cyber threat detection system intended to guard IoT-enabled systems against different forms of cyber threats.

Cybersecurity refers to a system that protects cyberspace from cyber threats and an intrusion detection system (IDS) falls under the umbrella of cybersecurity [1]. IDS can protect Internet-connected intelligent systems from both threats and cybercriminals. The functionality of IDS is to detect undesirable activity in incoming packets' payloads in the network traffic data and quarantine them from normal instances, and raise alerts [2]. IDS is classified into (i) technology type IDS and (ii) detection type IDS [11]. In detail, technology type IDS is further categorized as (i) host-based and (ii) network-based. In a host-based, computer activity such as configuration or log files of a particular operating system is tracked over an extended period. In contrast, network-based IDS can capture and analyze normal as well as real-world network attacks and synthetic attacks of packets inside the network. Moreover, detection type IDS is separated into (i) misuse-based (*MB-IDS*), and (ii) anomaly-based (*AB-IDS*). In *MB-IDS*, patterns such as the sequence of bytes are matched and then compared with the database, which is also known as signature-based IDS. *MB-IDS* can inspect packet information deeply and perform better in terms of accuracy. However, it cannot detect contemporary attacks and has limited scalability and search capabilities [14]. While *AB-IDS* creates a baseline for network traffic using statistical analysis and machine learning techniques to ensure the packets are investigated accurately in *AB-IDS*. Whenever any deviation from normal patterns occurs, the *AB-IDS* generates alerts about known as well as contemporary cyber threats to inform the user about this. However, *AB-IDS* suffers from a high false positive rate and a long training period required to build a model [2]. Therefore, it is necessary to design a more efficient IDS in order to enhance the cybersecurity of IoT-enabled networks.

In order to design an intelligent and efficient cyber threat detection framework, there are different challenges observed such as analysis of big cybersecurity data, different types of threats, and high dimensionality of features in network traffic data [12,15]. Consequently, the high dimensionality of the feature set increases the training time for the machine learning model and the processing cost. The classifier also struggles to distinguish between benign and malignant activity in network traffic. In order to reduce training time for computation, space, and time complexity, redundant and irrelevant features can be removed. Therefore, feature selection is a crucial process to get a reliable solution in the intelligent cyber threat detection model [12]. Moreover, another challenge is that some threat classes have a small sample size. Therefore, it may be difficult for a single classifier to detect threats with the lowest rate of false positives while maintaining a maximum threat detection rate [12,14]. This is because classifiers uses particular kernel functions, for example, probability values or attribute values, as the basis for their classification [2]. So, to address this problem (i.e., high misclassification rate), the proposed framework designs the cyber threat detection model using ensemble learning-based classifiers. The main aim of an ensemble learning technique is to integrate the opinions of multiple classifiers in order to achieve better performance in cyber threat detection.

The most crucial challenges that need to be addressed are listed below:

(a) A cyber threat detection model that is designed without proper use of data pre-processing and feature selection could be less accurate, have a higher FPR, and be less successful in detecting cyber threats [16,17]. Hence, detecting cyber threats in IoT-enabled networks can be viewed as a difficult and challenging task that needs to analyze a variety of features to determine which features work best.

(b) Most of the existing cyber threat detection schemes suffer from the "curse of dimensionality". The "curse of dimensionality" occurs when the dimensionality of a threat signature is exceptionally high, which increases a model's computation time and complexity during learning latent features. Thus, the identification and selection of robust and appropriate features for ensuring correct and timely threat detection is a challenging research issue.

(c) The FPR of a cyber threat detection model is usually increased when accuracy is improved [14]. Therefore, to design an effective cyber threat detection model with an optimized subset of features as well as an appropriate selection of classification approach that will improve the accuracy (*ACC)*, precision (*PR*), F1-score, detection rate (*DR*) with the lowest FPR which is yet another challenging task.

Main Contribution: This research work proposes an intelligent cyber threat detection framework that detects cyber threats with an optimal number of features, maximum ACC, and minimum FPR in network traffic. The key contributions of the proposed framework are summarized as follows:

i. Proposed a metaheuristic-based ensemble feature selection framework by using the BGSA and BGWO techniques. The proposed feature selection method is named *GSGW-DT*. In addition, a novel fitness function is formulated to measure the quality of the solution.
ii. To analyze the massive network traffic data and being able to make rapid and efficient decisions in the context of IoT-based networks, Decision Tree (DT) and ensemble learning-based detection model such as AdaBoost (AB), and Random Forest (RF) are used to detect the cyber threats.
iii. Results evaluation of the proposed cyber-threats detection framework is compared with the latest state-of-the-art models for the same experimental scenarios using the UNSW-NB15 dataset in terms of the number of selected features, ACC, DR, PR, F1-Score, and FPR. In addition, a stratified 10-fold cross-validation resampling process is used to assess each classification algorithm.

In addition to this section, the paper has the following sections: In Section 2, a brief outline of existing literature related to cyber threat detection is reviewed. Section 3 presents an overview of feature selection, the gravitational search algorithm, the grey wolf optimization technique, and ensemble learning methods. Section 4 presents the working methodology of the proposed framework. In Section 5, the outcome of the experimental results is discussed. Next, the conclusion and future work are discussed in Section 6.

2. Literature review

In this section, state-of-the-art cyber threat detection mechanisms for IoT-enabled networks are reviewed. This study considers literature related to cyber threat detection techniques using feature selection, hybrid techniques, and ensemble learning.

Nazir et al. [14] have proposed a cyber-attack detection model named TS-RF. In this work, RF is used to detect threats in IoT networks. The proposed model minimizes the time complexity and features of the UNSW-NB-15 dataset. This model achieved 83.12% of ACC and 3.7% of FPR. However, a few other significant evaluation metrics such as, PR, DR, and F1-Score were not considered. The model also suffers from low ACC and high FPR. Alazzam et al. [18] propose a model in which the discrete optimization problem of feature selection can be handled by cosine similarity. The technique named *Cosine_PIO*. In

this technique, the continuous pigeon inspired optimizer with modified sigmoid function using decision tree are used to get the relevant vector of features. Their model reduced the count of features from 49 to only 5, 41 to 7, and 41 to 5 features for the *UNSW-NB-15*, KDDCUP-99, and NSL-KDD datasets respectively. In the model, 96% of ACC and 98.20% of DR were achieved. However, the model still suffers from high FPR.

In [19], authors have developed an Oppositional Crow Search Algorithm named OCSA for selecting optimal features. The model uses Recurrent Neural Networks for detection of Denial-of-Service threats. Binary classification is considered and the experimental result shows ACC (94.12%), and appropriate feature subset is 10 out of 41 for the KDDCUP-99 dataset. However, a crucial evaluation metric, the FPR is not discussed in this experiment. Also, the suggested approach was assessed using an outdated dataset, which lacks contemporary cyberattacks. Raman et al. [20] suggested an IDS based on SVM and improved BGSA. The experimental outcome demonstrates that the model achieved maximum detection rate with 42 to only 25 features for NSL-KDD dataset and 49 to only 30 features for UNSW-NB15 dataset. Kumar et al. [21] have introduced a rule-based IDS that utilized Information Gain for feature reduction. Integration of various decision tree models are applied to detect cyber threats. The effectiveness of the IDS is accessed by the UNSW-NB15 and RTNITP18 datasets. The outcome indicates that the model achieved the highest accuracy with 13 features out 47 features for the UNSW-NB15 dataset.

Alghanam et al. [1] suggested an enhanced feature selection model for NIDS named LS-PIO. According to the results, the ensemble of classifiers iForest, LoF, and OC-SVM achieved maximum ACC (99.82%) with 12 features out of 42 features for the KDDCUP-99 dataset. In addition, Zhou et al. [12] have anticipated an efficient framework for IDS. The technique is known as C*FS*-BA. The model uses the BAT algorithm to get optimized features. In addition, the model is constructed using voting-based ensemble classification using C4.5, RF, and Forest PA, which requires less computing time. The effectiveness of the framework is assessed by the NSL-KDD, AWID, and CIC-IDS2017 datasets. However, key evaluation metrics: PR and F1_score, are not considered in this experiment. In addition, Kumar et al. [16] designed an IDS by using fog computing to handle huge network traffic. A mutual gain-based filter method is used to minimize the count of features. After that to select the classifier, statistical techniques such as the correntropy measure are used. Then classifiers such as kNN, XG-Boost, and Gaussian Naïve Bayes are used to detect cyber threats. The effectiveness of the model is evaluated using UNSW-NB-15 and DS2OS datasets in terms of ACC, DR, FPR, PR, and F1-score. For the DS2OS dataset, the model achieved 99.41% ACC with 10 features out of 12. Moreover, for the UNSW-NB dataset, the model obtained 97.62 of ACC with 16 features out of 49. However, filter methods are used for feature selection whereas wrapper-based methods are more efficient to get the most appropriate subset of features.

Garg et al. [17] have suggested a model to mitigate irrelevant features and detection of threats in network traffic. The model is an ensemble scheme named En-ABC. In En-ABC, relevant features are selected by the Restricted Boltzmann Machine named the RBM technique. Then the impact of non-linear features is handled by the Unscented Kalman Filter technique. Further, the Artificial Bee Colony (ABC) technique along with Fuzzy C-Means are used for clustering. IBRL, NSL-KDD, and NAB datasets are used to evaluate the model's performance. Experimental result shows that En-ABC achieves the highest ACC of 97.62%, 98.92%, 99.98%, and 99.97% for NSL-KDD, NAB, IBRL, and Synthetic datasets respectively. However, this technique uses an outdated NSL-KDD dataset for its analysis. Further, the complexity and execution time of the model is not considered by the authors. Kumar et al. [22] have developed a deep learning-based hybrid model for cyber threat detection known as *BDTwin*. The model uses the AR-DAVE technique to select useful features from the datasets. Then cyberattacks are detected by the A-BLSTM technique. As indicated by experimental results, the model has reached 99.12% and 99.37% ACC, respectively, using the TON-IoT and CICIDS-2017 datasets. However, a detailed discussion of the selected features is not provided by the authors.

Kumar et al. [2] have proposed a hybrid model for cyber-attack detection. Feature selection is performed using the hybridization of filter methods. Feature vectors obtained by each method are combined using the logical AND operation to get a single reduced feature subset. Then, the correntropy measure is used to select the classifier. To detect cyber threats RF, XGBoost, and KNN are used. The effectiveness of the model is assessed by the DS2OS, NSL-KDD, and BoT-IoT datasets. The suggested model achieved 99% ACC. However, one of the most crucial evaluation metrics, FPR was not taken into account in this experiment. Hajisalem et al. [11] presented another hybrid framework named ABC-AFS. First, the fuzzy C-Means Clustering approach is used to divide the dataset into many subsets. Then a filter method such as Correlational Coefficient is applied to get a reduced set of features. The CART technique is then applied to detect cyber threats. The effectiveness of the framework is assessed by the NSL-KDD and UNSW-NB 15 datasets. The framework achieved 99% DR and 0.01% FPR on the NSL-KDD dataset. However, data pre-processing is not discussed in this work. Kumar et al. [23] have designed a distributed IDS to detect Distributed Denial of Service threats. In this framework, cyber threats are detected by Random Forest and XGBoost. The BoT-IoT dataset evaluates the effectiveness of the framework. This technique achieved ACC of 99.99% and 99.98% for the binary-class and multi-class detection problems respectively. However, overall FPR was not discussed in this experiment.

Tama et al. [24] have suggested another hybrid model named *TES-IDS*. Metaheuristic techniques such as ACO, GA, and PSO are adapted for feature selection. Then an innovative two-stage classifier is designed to detect cyber threats. The effectiveness of the model is assessed by the UNSW-NB 15 and NSL-KDD dataset. Experimental results indicate that model achieved maximin ACC of 91.27% and minimum 19 features out of 42 in the UNSW-NB15 dataset. However, result shows low DR for R2L and U2R threats and the model still suffer from high FPR. In addition, Dwivedi et al. [25] have presented a novel hybrid technique named EFSGOA to detect intrusions in network traffic data. Optimal features are selected by filter methods such as CMIM, mRMR, and JMI. After that, to get final subset of features, grasshopper optimization technique is applied. Then benign and malicious activities are detected by SVM. Based on the experiment result model achieved ACC of 99.98% and 99.89% for the NSL-KDD and KDDCUP-99 datasets respectively. This approach, however, relies on an outdated dataset that does not incorporate advanced cyber threats. Kumar et al. [26] have proposed a deep learning-based technique named *PBDL*. In this technique, data is secured by blockchain technology then a novel hybrid deep learning technique named SA-BiLSTM is used to detect cyber threats. The IoT-Botnet and ToN-IoT datasets are used to evaluate the performance of the technique. However, overall DR and FPR were not discussed in this experiment. Kumar et al. [27] have proposed a blockchain and deep learning-based IDS model named *P2TIF*. The model uses blockchain techniques for secure communication and a deep variational encoder for data encoding. In addition, GRU is used to extract useful features. To detect cyber threats, an attention-based deep-gated recurrent neural network is used. Based on the experiment result, the model achieved ACC of 99.88% and 99.29% for the ToN-IoT and IoT-Botnet datasets respectively. However, the number of selected features, overall DR, and FPR is not discussed by the authors.

Oseni et al. [28] have developed a explainable framework using deep learning for cyber threat detection in IoT-enabled networks. Correlation Coefficient method to select the features in the framework. Convolutional neural network is applied to detect threats. The effectiveness of the framework is evaluated by ToN-IoT dataset. This technique achieved ACC of 99.15% and 90.55% for the binary-label and multi-label detection problem respectively. However, overall FPR was not discussed in this experiment. Kumar et al. [29] have developed a

Table 1
A brief overview of the literature on cybersecurity threats detection techniques for IoT-enabled networks.

References	Year	Detection problem	FS Technique	Hybrid Model	Datasets	No. of Selected feature	Detection technique	ACC (%)	DR (%)	FPR (%)
Hajisalem et al. [11]	2018	Binary and Multi-class	FCM, CFS	✓	NSL-KDD, UNSW-NB 15	6/41, 6/49	(ABC + AFS)	99.00, 98.90	99.00, 98.60	0.01, 0.13
Garg et al. [17]	2019	Multi-class	RBM, UKF	✓	NSL-KDD, NAB, IBRL, Synthetic	NA	(ABC + FCM)	97.62, 98.92, 99.98, 99.97	97.59, 97.27, 95.48, 95.48	1.05, 1.13, 1.79, 0.99
Raman et al. [20]	2019	Binary	HC-IBGSA SVM	×	UNSW- NB15, NSL-KDD	30/49, 25/42	Modified K means-SVM-ELM	94.11, 8.85	98.47, 98.72	2.18, 1.27
Kumar et al. [21]	2019	Multi-class	Information Gain	×	UNSW-NB15	13/47	Rule-based model	84.83	90.32	2.01
Tama et al. [24]	2019	Binary	TES-IDS	✓	UNSW- NB15, NSL-KDD	19/42, 37/43	Two-stage meta classifier	91.27, 85.79	91.30, 86.8	8.9, 11.7
Zhou *et al.* [12]	2020	Binary	CFS-BA	✓	NSL-KDD, AWID, CIC-IDS2017	10/41, 8/84, 13/78	Voting Ensemble (C4.5 + RF + Forest PA)	99.81, 99.52, 99.89	99.80, 99.50, 99.90	0.08, 0.15, 0.12
Nazir et al. [14]	2020	Binary and Multi-class	TS-RF	×	UNSW-NB 15	16/49	RF	83.12	NA	3.7
Kumar et al. [16]	2020	Multi-class	Mutual Information	✓	UNSW- NB15, D2OS	16/49, 10/12	RF	97.62, 9.41	97.59, 99.99	1.05, NA
Alazzam et al. [18]	2020	Binary	PIO	×	KDDCUP-99, NSL-KDD, UNSW-NB15	7/41, 5/41, 5/49	DT	96.0, 83.30, 91.70	98.20, 86.60, 89.40	7.6, 8.8, 3.4
SaiSindhuTheja et al. [19]	2020	Binary	OCSA	✓	KDDCUP-99	10/41	RNN	94.12	95.13	NA
Kumar et al. [2]	2021	Multi-class	CC, RFDC, GR	✓	NSL-KDD, DS2OS, BoT-IoT	18/41, 6/12, 10/19	XGBoost	99.34, 99.43, 99.99	98.67, 95.65, 98.56	NA
Kumar et al. [29]	2021	Multi-class	NA	×	ToN-IoT	NA	DFE-CTIDD DFE-CTIATI	99.98 99.33	NA	NA
Dwivedi et al. [25]	2021	Multi-class	mRMR, JMI, CMIM, GOA	×	KDDCUP-99, NSL-KDD	6/41, 5/41	SVM	99.89, 99.98	99.26, 99.69	0.097, 0.07
Kumar et al. [27]	2022	Multi-class	GRU	×	ToN-IoT BoT-IoT	×	A-DGRNN	99.88 99.29	NA	NA
Kumar et al. [22]	2022	Multi-class	AR-DAVE	✓	ToN-IoT CICIDS-2017	×	A-BLSTM	99.12 99.37	NA	NA
Alghanam et al. [1]	2022	Binary	LS-PIO	×	UNSW-NB15, -KDDCUP-99, NSL-KDD, BoT-IoT	8/49, 12/42, 10/42, 03/19	Ensemble (iForest + LoF + OC-SVM)	94.45, 99.82, 94.7, 97.37	96.7, 99.7, 97.5, 99.7	11.1, 2.4, 11.2, 1.02
Kumar et al. [23]	2022	Binary and Multi-class	NA	×	BoT-IoT	×	XGBoost RF	99.99 99.98	99.99 99.99	NA
Kumar et al. [26]	2022	Multi-class	NA	✓	IoT-Botnet ToN-IoT	×	SA-BiLSTM	99.98 99.89	NA	NA
Oseni et al. [28]	2022	Binary and Multi-class	CC	×	ToN_IoT	NA	CNN	99.15, 90.55	99.15, 90.55	NA
Dey et al. [30]	2023	Binary	NSGA-II	✓	ToN_IoT	13/42	SVM	99.48	NA	NA

Abbreviations: NA indicates no information available.

cyber threat detection model named *DLTIF*. In this model, deep feature extractor is used to find hidden pattern. Next, cyber threat intelligence is applied to detect cyber threats. Based on the experiment result, the model achieved the highest ACC of 99.98% for the ToN-IoT dataset. However, feature selection, overall DR, and FPR were not discussed in this experiment. Dey et al. [30] have proposed a hybrid framework for cyber threat detection. In this framework, features are selected by hybrid NSGA-II technique. Next, Support Vector Machine is used to detect cyber threats. Experimental results indicate the model achieved an ACC of 99.48% and selected 13 features from 45 in the ToN-IoT dataset. However, the authors have not discussed DR and FPR (see Table 1).

3. Preliminaries

In this section, a brief review of feature selection techniques, metaheuristics-based wrapper feature selection techniques such as: Gravitational Search Algorithm (GSA), Grey Wolf Optimization (GWO), and ensemble learning methods are discussed.

3.1. Feature selection (FS)

Feature selection finds the most relevant subset of features. In order to select features, four steps must be followed (i) generating the subsets; (ii) evaluating the subsets; (iii) testing the stopping criterion, and (iv) validating the results [14]. The primary goal of the first step is to

generate the subset which is a search process that finds the optimal feature subset from the dataset according to a certain criterion. In addition, in this step irrelevant and redundant features are eliminated, where irrelevant features refer to those that will not contribute to the classification process, and redundant features are those that are highly correlated. In the second step, each subset is evaluated by an objective/fitness function to choose the most relevant one. In the third step, iterations are performed until a predefined stopping condition is reached. Finally, results are validated by performance metrics.

FS technique is categorized into three types such as filter-based, wrapper-based, and embedded methods [11]. Filter-based FS determines the relevance of features in a dataset and selects the most relevant features based on statistical analysis [12]. In the wrapper method, the search strategy chooses the subset of features to be assessed. After selecting features, any classification technique is used to assess their performance against the objective function [14]. In contrast to wrapper method, embedded method selects features during the learning process and it is less computationally expensive but modification is quite difficult in the classification model to obtain better performance [12]. In the field of cybersecurity, the main reason to apply FS is to improve the detection accuracy of the learned models. In addition, to reduce the computational cost of the data and the time-space complexity of the learned models. Therefore, in this work, a wrapper-based *FS* technique is adapted to get a more optimized subset of features because the wrapper method provides better results than the filter method [14].

3.2. Overview of GSA

GSA that solves complex optimization problems. GSA is based on Newton's law of gravity which says "Every particle with mass in the universe attracts another particle with a gravitational force" [31]. In GSA-based feature selection, each particle is considered an object (or agent) referred to as a possible solution to the optimization problem. An object's performance is calculated by its mass, which is described in terms of the fitness function. In general, an object having a heavier mass (min or max fitness value) is considered as a better solution. This factor attracts the other objects towards the better solution with a gravitational force. In the initial phase, each object is assigned a random mass and position. After initialization, each object's mass is adjusted by updating its position. The step-by-step description of the GSA algorithm [31] can be given as follows:

Step 1 ***Initialization:*** Suppose k objects are initialized with random positions. In an n-dimensional space, ith object position can be defined as in Eq. (1):

$$P_i(t) = \left(P_i^1, P_i^2, \cdots \cdots P_i^d \cdots \cdots \cdots, P_i^n\right); \forall i = 1, 2, \cdots \cdots \cdots \cdots k \tag{1}$$

Where $P_i(t)$ represents the position vector of ith object at a specific time t called iteration, value of each vector represents the position of object in that dimension, for example P_i^d is position of ith object in d dimension.

Step 2 ***Mass calculation:*** For each object called feature subset, mass is calculated by fitness function as defined in Eq. (23). The mass $W_i(t)$ for ith object at time t is calculated on the basis of fitness function ($f_i(t)$) of that object at a particular time (iteration) t as defined in Eq. (2) and Eq. (3).

$$W_i(t) = \frac{f_i(t)}{\sum_{j=1}^{k} f_j(t)}; \forall i = 1, 2, \cdots \cdots \cdots k \tag{2}$$

$$f_i(t) = \frac{fitness_i(t) - worst(t)}{best(t) - worst(t)} \tag{3}$$

Where $fitness_i(t)$ is fitness value of ith object at time (iteration) t by fitness function. For maximization problem, $best(t)$, $worst(t)$ are described in Eq. (4) and Eq. (5). In the present work, the cyber threat detection problem is regarded as a maximization problem i.e., Eqs. (4) and (5) are used.

$$best(t) = \underset{j\in\{1,\cdots,n\}}{maximum} fitness_j(t) \tag{4}$$

$$worst(t) = \underset{j\in\{1,\cdots,n\}}{minimum} fitness_j(t) \tag{5}$$

Step 3 ***Gravitational force computation:*** Gravitational force is an attractive force applied between two masses of objects, separated by distance as defined in Eq. (6).

$$F_{ij}^d(t) = G(t)\left(\frac{W_{pi}(t)\cdot W_{aj}(t)}{D_{ij}(t)+\varepsilon}\cdot\left(P_j^d(t) - P_i^d(t)\right)\right) \tag{6}$$

Where $F_{ij}^d(t)$ represents the force between mass of ith and jth object in dimensional d at time t called iteration, $W_{pi}(t)$ represents passive gravitational mass of ith object at time t, $W_{aj}(t)$ represents active gravitational mass of jth object at time t, $P_j^d(t)$ represents the position of jth object at time t, $P_i^d(t)$ represents the position of ith object at time t, ε is used as a trivial constant, $G(t)$ represents the gravitational constant at time t as defined in Eq. (7).

$$G(t) = G_0 * exp\left(\frac{-a * iteration}{iteration_{max}}\right) \tag{7}$$

Where G_0 is an initial gravitational constant, a is a decreasing coefficient, and $iteration_{max}$ represents maximum number of iterations.

$D_{ij}(t)$ represents the Euclidian distance between ith and jth object at time t as defined in (8).

$$D_{ij}(t) = \left\|P_i(t), P_j(t)\right\|_2 \tag{8}$$

Where $P_i(t)$ represents position of ith object at time t, and $P_j(t)$ represents position of jth object at time t.

In order to provide a stochastic experience to GSA, the authors [31] believe that at time t and dimension d, the total force that acts on object i is a random-weighted sum of cth components of the forces applied by k_{best} object, as defined in Eq. (9).

$$F_i^d(t) = \sum_{j=k_{best}, j\neq i}^{k} r_j F_{ij}^c(t) \tag{9}$$

Where $F_i^d(t)$ is total force that acts on ith object in a dimension d at time (iteration) t, r_j represents uniform random value in interval [0,1], and k_{best} represents the first k objects with greatest fitness value. It should be noted that only k_{best} objects will attract other objects i.e., a set of k_{best} objects begin with all the objects and drops linearly to one after each iteration, so this is a constraint in the GSA. As a result of this constraint, k_{best} objects will make better the efficiency of the GSA by monitoring exploitation and exploration to prevent being trapped in local optimum solutions [31].

Stage 4 ***Acceleration:*** In the next step, acceleration is calculated on the basis of the law of motion, as defined in Eq. (10).

$$A_i^d(t) = \frac{F_i^d(t)}{W_{ii}(t)} \tag{10}$$

Where $A_i^d(t)$ represents the acceleration of ith object in dimension d at time t, $F_i^d(t)$ is total force as defined in Eq. (9), $W_{ii}(t)$ represents the inertia mass of ith object at time t.

Stage 5 ***Velocity and Position:*** Then, updated velocity and position of each object is calculated using Eqs. (11) and (12).

$$V_i^d(t+1) = r_i \cdot V(t) + A_i^d(t) \tag{11}$$

$$P_i^d(t+1) = P_i^d(t) + V_i^d(t+1) \tag{12}$$

Where $V_i^d(t+1)$ represents updated velocity of ith object in dimension d at time $t+1$, r_i represents uniform random value in interval [0,1], $V_i^d(t)$ represents current velocity of ith object in dimension d at time t, and $A_i^d(t)$ represents the acceleration as defined in Eq. (10), $P_i^d(t+1)$ represents updated position of ith object in dimension d at

f_1	f_2	f_3	f_4	…	…	f_{n-1}	f_n
1	1	0	1	…	…	0	1

Fig. 1. Solution representation.

time $t+1$, and $P_i^d(t)$ position of ith object in dimension d at time t. However, one can observed that Eq. (12) cannot be applied in this work, as feature selection is a discrete optimization problem [18] and GSA can used to solve continuous optimization problem. Therefore, the BGSA is proposed by the authors [32] as discussed in Section 3.3.

3.2.1. Overview of BGSA

BGSA is the process of converting the continuous solution obtained by the GSA to a binary solution [32]. As discussed in Section 3.2, GSA can solve continuous optimization problem and feature selection is a discrete optimization problem [18]. Hence, it is necessary to convert the solution of GSA into the binary form to represent the FS problem [33]. Fig. 1 illustrates how a random binary value can represent the solution, in which features that are selected are represented as 1 and those that are not selected are represented as 0. However, the BGSA use the same computational steps as GSA, which is discussed in Section 3.2 but the only difference between GSA and BGSA is that the position of each object is updated by Eqs. (13) and (14) instead of Eq. (12) in BGSA.

$$P_i^d(t+1) = \begin{cases} 1 - P_i^d(t); & r_i < Sf\left(V_i^d(t+1)\right) \\ P_i^d(t); & otherwise \end{cases} \tag{13}$$

Such that

$$Sf(x) = \frac{1}{1+e^{-x}} \tag{14}$$

Where $P_i^d(t)$ is position of ith object in d dimension at cycle t, r_i represents random value in interval [0,1], $V_i^d(t)$ is velocity of ith object in d dimension at cycle t, $Sf(x)$ is sigmoid function for binary transformation and e represents Euler value.

3.3. Overview of GWO

GWO is a population-based metaheuristic algorithm based on swarm intelligence that impersonates the social hierarchical leadership structure and hunting mechanism of grey wolves in nature [34]. The GWO used four types of grey wolves: alpha (α), beta (β), delta (δ), and omega (ω) to simulate a hierarchical leadership structure. Grey wolves usually prefer to live in a group. The average group size of grey wolves is between 5 and 12 individuals. The process of forming a grey wolf group involves four stages. First, the highest wolf and leader of the group is called α. The α wolf's responsibilities include hunting, picking a sleeping area, and deciding when to walk. The second highest wolf in the group is β, which helps α to make decisions, reinforcement, and other activities. It will be lead if α is dead or sick. The third highest wolf is δ, reporting to the α and β wolves, and is responsible for governing the ω wolf. The δ wolf provides alerts about any threat, or wounded wolves to the group, and cares for the weakest members of the group. Finally, the weakest or lowest level wolf in the group is ω. As part of the hierarchical system, the omega wolves are subordinate to the top-tier wolves. Their role in the group is to act as victims when in danger. Omega wolves are the last to eat in a group. The absence of the omega wolf has led to violence among all wolves. There is a perception that omega wolves are the babysitters in the group. There is a mathematical model in GWO that represents social hierarchy, encircling prey, hunting prey, and attacking prey at various stages. Details of each stage are defined as follows:

(a) Social Hierarchy Alpha (α) wolf represents the best possible solution. β and δ wolves are the second and third most effective possible solutions. The Omega (ω) wolves are thought to be the remaining possible solution. The optimized solution is obtained by α, β, δ, and a group of ω wolves follow the rest of the wolves.

(b) Encircling Prey: A wolf in the encircling stage locates its prey and instructs all the other wolves to encircle the prey. Due to this, other wolves are required to update their locations in accordance with the instructions. Prey encircling process is mathematically represented in Eqs. (15) and (16):

$$\vec{D} = \left|\vec{V} \cdot \vec{Pp}(t) - \vec{W} \cdot \vec{P}(t)\right| \tag{15}$$

$$\vec{P}(t+1) = \vec{Pp}(t) - \vec{V} \cdot \vec{D} \tag{16}$$

Where t is current iteration, $\vec{V}$ and $\vec{W}$ are the coefficient vectors, $\vec{Pp}$ is position vector of prey, and $\vec{P}$ represents the position vector of grey wolf. Calculation of random vector $\vec{V}$ and $\vec{W}$ that are describe in Eqs. (17) and (18)

$$\vec{V} = 2\vec{a} \cdot \vec{rand_1} \cdot \vec{a} \tag{17}$$

$$\vec{W} = 2 \cdot \vec{rand_2} \tag{18}$$

Where $\vec{a}$ is a parameter, which is decreased by 2 to 0 in each cycle and $\vec{rand}_1$ and $\vec{rand}_2$ represents random vectors in $[0,1]$.

(c) Process of Hunting: In the hunting process, α supervises the β and δ. α represents the fitness solution or most suitable position. β and δ are in second and third-best position respectively. ω wolves bring up to date their positions based on α, β, and δ. The mathematical formula of Eqs. (19)–(21) for the hunting process is as follows:

$$\left.\begin{aligned} \vec{D_\alpha} &= \left|\vec{W_1} \cdot \vec{P_\alpha} - \vec{P}\right|, \\ \vec{D_\beta} &= \left|\vec{W_2} \cdot \vec{P_\beta} - \vec{P}\right|, \\ \vec{D_\delta} &= \left|\vec{W_3} \cdot \vec{P_\delta} - \vec{P}\right| \end{aligned}\right\} \tag{19}$$

Where $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ are distance vector for α, β, and δ respectively.

$$\left.\begin{aligned} \vec{P_1} &= \vec{P_\alpha} - \vec{V_1} \cdot \vec{D_\alpha}, \\ \vec{P_2} &= \vec{P_\beta} - \vec{V_2} \cdot \vec{D_\beta}, \\ \vec{P_3} &= \vec{P_\delta} - \vec{V_3} \cdot \vec{D_\delta} \end{aligned}\right\} \tag{20}$$

$$\vec{P}(t+1) = \frac{\vec{P_1} + \vec{P_2} + \vec{P_3}}{3} \tag{21}$$

Where $\vec{P}_1$, $\vec{P}_2$, and $\vec{P}_3$ represents first three solutions of population. However, one can observed that Eq. (21) cannot be applied in this work, as feature selection is a discrete optimization problem [18] and GWO can used to solve continuous optimization problem. Therefore, BGWO is proposed by authors [35] as discussed in Section 3.5.

(c) Attacking Prey or Searching Prey: Grey wolves attack their prey only when they have ceased to move. In attacking prey i.e., exploitation, $|A| < 1$, the wolves are forced to attack their prey. Throughout the iteration, a is decreased from 2 to 0 at random value $\vec{A}$ within interval [2a, −2a]. In searching for prey i.e., exploration, each solution (wolf) updates its location according to the location of α, β, and δ. When $|A| > 1$ or $|A| < -1$ then diverse from the prey and find the better.

3.3.1. Overview of BGWO

BGWO is the process of converting the continuous solution obtained by *GWO* to a binary solution [35]. As discussed in Section 3.3, *GWO* can tackle continuous optimization problems but the selection of features is a discrete optimization problem [18]. Hence, it is necessary the solution of GWO should be converted into binary values {0,1} in nature. Fig. 1 illustrates how a random binary value can represent the BGWO solution. However, BGWO follows the same computational steps as discussed in Section 3.3 for GWO, but the only difference between

BGWO and GWO is that the position of the wolves is computed by Eq. (22) instead of Eq. (21) in BGWO.

$$\mathcal{P}_d^{t+1} = \begin{cases} 1, & if\ Sf\left(\frac{\overrightarrow{P_1}+\overrightarrow{P_2}+\overrightarrow{P_3}}{3}\right) \geq r \\ 0, & otherwise \end{cases} \tag{22}$$

Where $\mathcal{P}_d^{t+1}$ updated binary location at cycle t in dimension d, r is a random number between [0,1], $Sf(x)$ is sigmoid function for binary transformation and e represents Euler value as defined in Eq. (14), $\overrightarrow{P_1}, \overrightarrow{P_2}, \overrightarrow{P_3}$ are position vectors as defined in Eq. (19).

3.4. Formulation of fitness function

In an optimization problem, a fitness function measures the quality of the solution [36]. Its purpose is to determine what is the most suitable solution in a population. In the metaheuristic-based wrapper *FS* problem, formulation of the fitness function is extremely important [33], as it is calculated for each possible solution within a search space. In the proposed work, a new fitness function is formulated, as defined in Eq. (23), which evaluates the most suitable solution on the basis of two objectives: (i) Selected feature subset using random population initialization and (ii) Based on selected feature subset, classification accuracy of a reduced train set. For each objective, Eq. (23) is used for calculating the fitness value which is defined as follows:

$$ff = (w_1 * acc) + (w_2 * \frac{1}{sf}) \tag{23}$$

Where ff represents fitness function, acc represents classification accuracy of a reduced train set, sf represents number of selected features, and $w_1 + w_2 = 1$. In order to ensure equal weights for acc and sf, the weights are set as $w_1 = w_2 = 0.5$.

3.5. Ensemble learning methods for cyber threat detection

Ensemble learning methods are machine learning models that combine the opinions of multiple learners to achieve better performance [1]. It is classified into three categories: Boosting, Bagging, and Stacking techniques. The procedure of boosting-based ensemble learning method is to train multiple homogeneous learners sequentially and combines them for prediction, where every learning model aims to fix the errors made in the previous learner [12]. In bagging, random samples from the training set, which is called bootstrap, are selected to train the multiple homogeneous learners parallelly and combine them for prediction [12]. In both boosting and bagging, multiple single homogeneous learners are clubbed together to achieve higher performance. While stacking-based ensemble methods differ from bagging and boosting in two aspects: (i) individual learners may be heterogeneous; (ii) meta-learner can be any model. The procedure is that the heterogeneous learners will have some prediction and this becomes input to the meta-learner model. Next, whatever the final prediction will get from the meta-learner model that becomes the final prediction but the time complexity of the stacking-based ensemble method is quite high as compared to boosting and bagging-based ensemble methods [37].

4. Proposed cyber threats detection framework for IoT-enabled networks

In this section, a detailed description of the working idea of the proposed cyber threat detection framework is presented. Figs. 2 and 3 illustrate the flow diagram of the proposed framework. The proposed framework is designed based on the concept of anomaly-based NIDS for the detection of cyber threats. The proposed framework consists of three phases: data pre-processing phase, feature selection phase and threat detection phase. Detail explanation of each phase is discussed below.

(a) **Phase 1: Data Pre-processing** This section describes the steps involved in data pre-processing. As IoT-enabled networks generate traffic from several devices and sensors, it is observed that these traffics contains various types of features including categorical and symbolic data. For this reason, data pre-processing is crucial to the creation of a robust and effective detection model, as described in the following.

i. **Label and Data Transformation**: Symbolic and categorical data from IoT network traffic are transformed into numerical numbers through data and label transformations. In addition, the target class is converted into binary classes 1 or 0, where 1 refers to normal traffic whereas 0 refers to malicious traffic.

ii. **Normalization**: It has been observed that features in IoT network traffic have distinct magnitude values, which can slow the learning process. In this model, min–max data normalization is applied in the dataset to rescaling the data value of each feature into the range [0,1]. This is because data normalization is a crucial step to avoid bias of higher values features from the dataset [16]. Also, to enhance convergence, and reduce training time, data normalization is an important step. The normalization formula [2] is applied by using Eq. (24).

$$F_{normalized} = \frac{F - F_{minimum}}{F_{maximum} - F_{minimum}} \tag{24}$$

Where F is the attribute that should be rescaled, $F_{minimum}$ represents minimum value, and $F_{maximum}$ represents maximum value for each attribute in the dataset.

(b) **Phase 2**: **Ensemble Feature Selection by Metaheuristic-based Wrapper Techniques:** In this phase, two metaheuristics-based wrapper feature selection techniques are designed in which Binary Gravitational Search Algorithm (BGSA) and Binary Grey Wolf Optimization (BGWO) are used as feature optimizers and Decision Tree (*DT*) is used as its evaluator for cyber threat detection in IoT-enabled networks. The designed technique is based on B**GS**A and B**GW**O metaheuristic as FS optimization and DT is used as evaluator. Thus, the designed techniques are referred to as *GS-DT* and *GW-DT,* respectively.

Firstly, the reason to apply metaheuristic-based wrapper FS techniques is because these techniques are more efficient to tackle the search space complexity. Search space complexity means suppose network traffic data contains 'n' number of features then possible count of proper subset of features will be 2^n-1. This implies that the search space complexity of various subsets of features will have $2^n - 1$ [38], which belongs to the NP-hard problem [14] that can be handled in an efficient manner by the metaheuristic-based wrapper techniques. In addition, wrapper techniques are more accurate than filter methods [14]. Secondly, a significant advantage of metaheuristic-based wrapper methods is that they use classifiers to guide the feature selection process [35], since a classifier evaluates features based on their predictive accuracy. In this model DT classifier is used to train and evaluate the features subset. The reason to use DT as a class evaluator in BGSA and BGWO is because DT is able to handle complex features interrelation more easily than other types of base classifier [18,39]. After that, we define a novel fitness function as defined in Eq. (23) is used to get the most suitable subset of features, and the workflow of this process is shown in Fig. 3.

The next step is to design an ensemble of *GS-DT* and *GW-DT* techniques to obtain more optimal features, which is referred to as a *GSGW-DT* model. This ensemble model integrates and takes advantage of *GS-DT* and *GW-DT* techniques. As a result of this techniques, two feature subsets are obtained. The first subset of features is selected by the *GS-DT* denoted by which is $S_{opt\_features1}$. The second subset of features is selected by the *GW-DT* denoted by which is $S_{opt\_features2}$. In the next step, the set intersection ($\cap$) operation is applied on the above two subsets to get a single and more optimized subset of features (*i.e.*, all the features are selected from the above two subsets if and only if features are common), as defined in line # 12 of Algorithm 1. Then the final resulting subset of features obtained by *GSGW-DT* which is denoted by

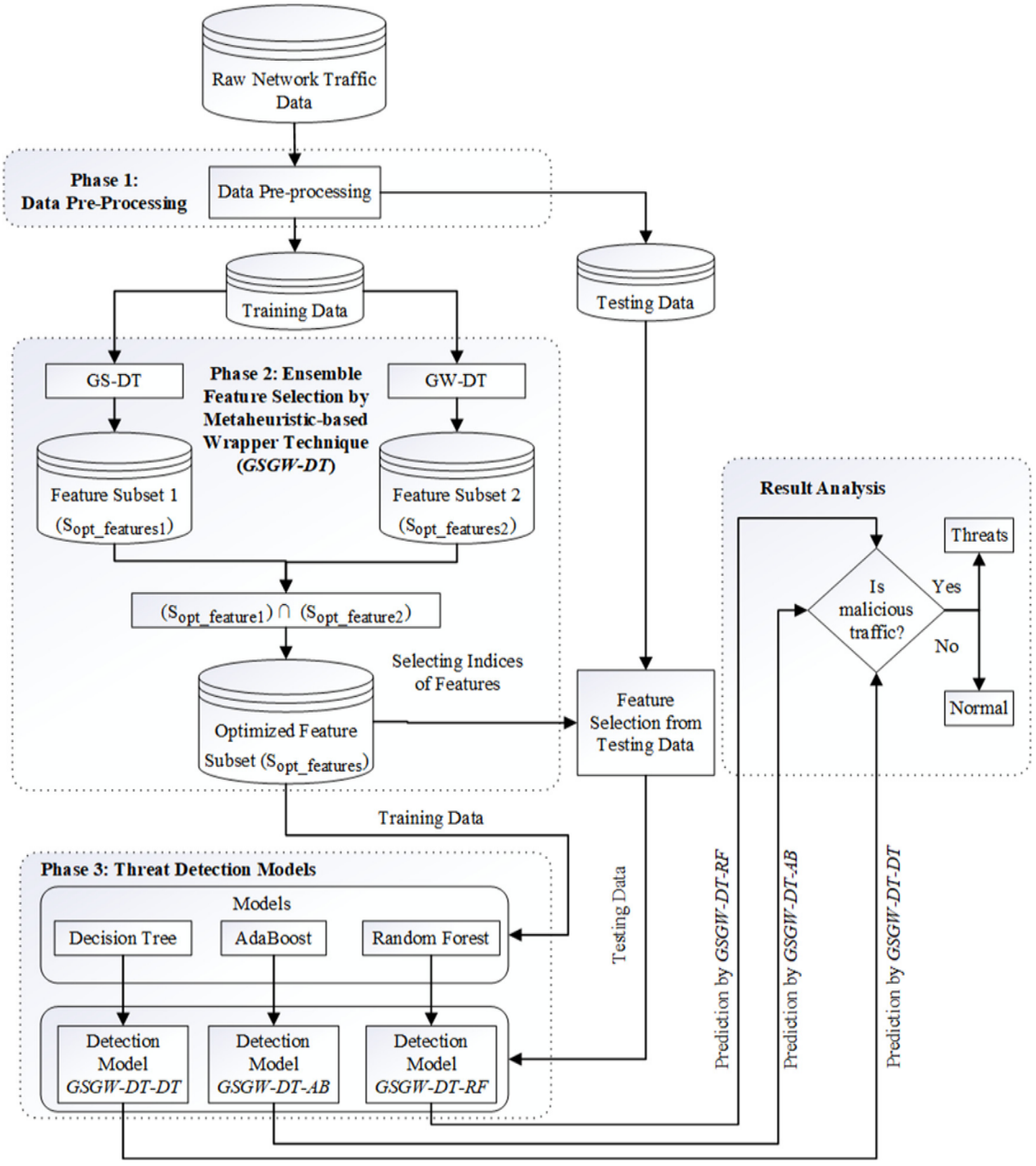

Fig. 2. The overall workflow of the proposed framework for cyber threat detection.

$S_{opt\_features}$ is fed into the DT classifier as well as ensemble learning-based classifiers such as AdaBoost and Random Forest separately to detect cyber threats in IoT-enabled networks. Algorithm 1 shows the proposed ensemble feature selection technique.

4.1. Phase 3: Cyber threat detection using machine learning algorithms

In this phase, three renowned machine learning approaches, such as DT, AdaBoost, and RF, are applied separately to the optimized feature subset ($S_{opt\_features}$) which is obtained by the *GSGW-DT* model. The designed detection model based on DT, AdaBoost, and Random Forest, which are referred to as *GSGW-DT-DT*, *GSGW-DT-AB*, and *GSGW-DT-RF* respectively. Here, AdaBoost and Random Forest are ensemble learning-based classification techniques. This is because many researchers stated that the performance of ensemble learning-based classifiers are much better than the use of a single classification technique [12]. A brief description of the machine learning algorithms used in this research work is presented below:

(a) **Decision Tree (DT)**: The approach used in the DT classifier is a top-down, recursive, and divide-and-conquer approach. A greedy approach is used to solve a classification problem. The main procedure is to select a best feature (i.e., root node) and split it into from a larger training set into smaller training sets according to a certain splitting criteria [33]. In the DT model, classification problem can be solved by asking the series of questions (i.e., the values) about the independent features from the training dataset. In which the independent features and their dependent features (i.e., target features) are organized in the form of decision tree. Fig. 4 shows the hierarchical structure of the DT consisting of root node, internal nodes, leaf nodes, and edges where root node and internal nodes represents independent features, edges are the values of the independent features, and leaf node represents the class-label i.e., outcome of the classification. Once a DT model is built, it is applied to the test dataset for classification.

(b) AdaBoost: The AdaBoost algorithm, also called Adaptive Boosting, is a repetitive method of boosting to classify the binary class [40]. It works by selecting and combining base learners in order to create a

Algorithm 1: Ensemble feature selection technique: *GSGW-DT*

Input: Set of actual features $f \in \{f_1, f_2, f_3, \cdots\cdots\cdots, f_n\}$

Output: Optimized subset of features ($S_{opt\_features}$)

1. Initialize random population of k objects in n-dimension feature space $f \in \{f_1, f_2, f_3, \cdots\cdots\cdots, f_n\}$
2. Calculation to get first optimized feature subset $S_{opt\_features\,1}$ using *GS-DT*
3. **for** index $= 1, 2, 3, \cdots\cdots\cdots, n)$ **do** // n is total number of features
4. Compute GS-DT by using BGSA
5. $S_{opt\_features\,1} = f \in \{f_1, f_2, f_3, \cdots\cdots\cdots, f_p\}$ //where p represents the optimized features in subset $S_{opt\_features\,1}$
6. **end for**
7. Calculation to get second optimized feature subset $S_{opt\_features\,2}$ using *GW-DT*
8. **for** index $= 1, 2, 3, \cdots\cdots\cdots, n)$ **do**
9. Compute GW-DT by using BGWO
10. $S_{opt\_features\,2} = f \in \{f_1, f_2, f_3, \cdots\cdots\cdots, f_q\}$ // q represents the optimized features in subset $S_{opt\_features\,2}$
11. **end for**
12. $S_{opt\_features} = \{f : f \in (S_{opt\_features\,1} \cap S_{opt\_features\,2})\}$
13. **return** $S_{opt\_features}$

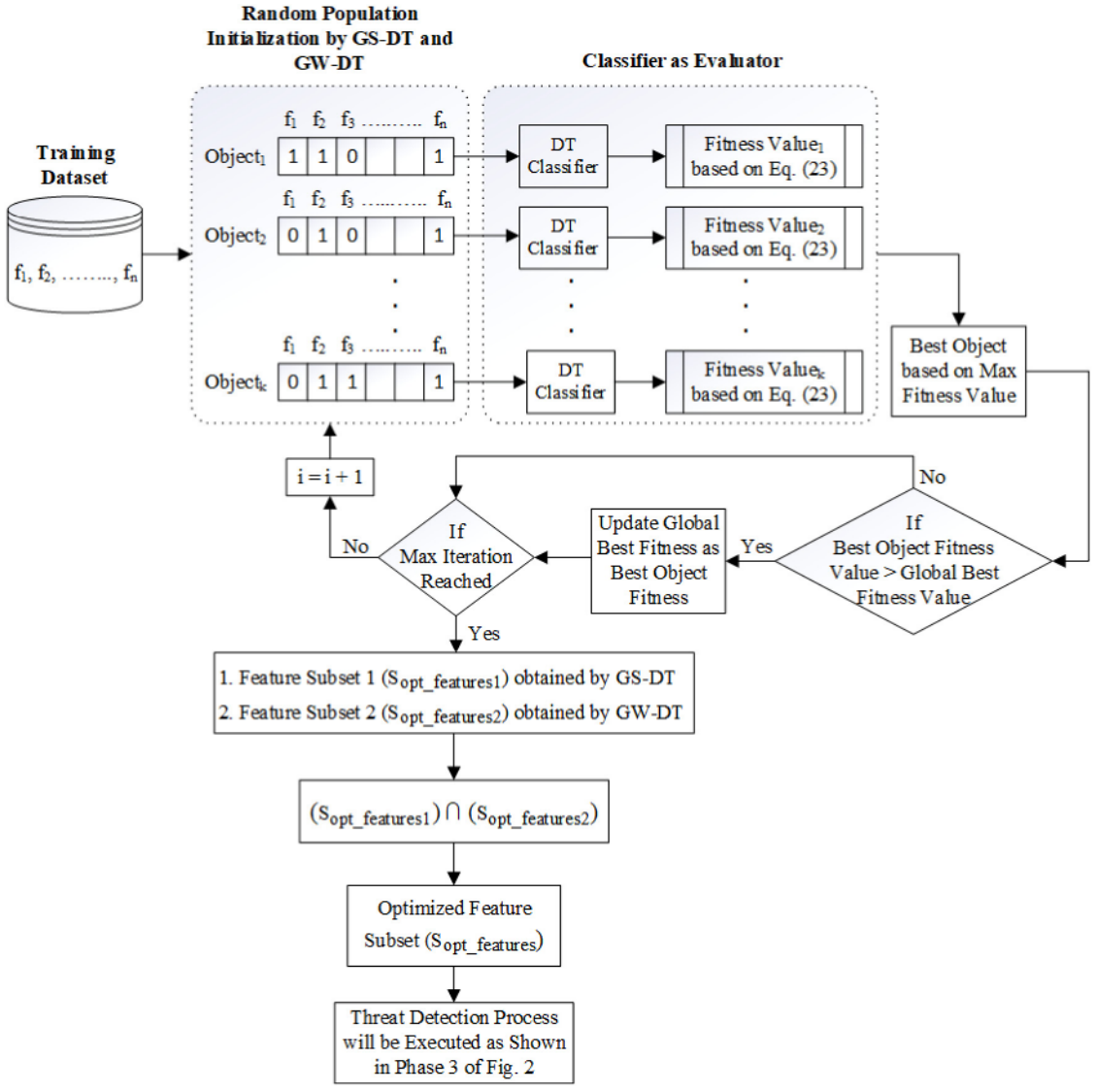

Fig. 3. Phase 2 workflow: Ensemble feature selection by metaheuristic-based wrapper technique (*GSGW-DT*).

strong classifier. Fig. 5 shows the implementation of AdaBoost in which initial base learners such as $model_1$ is selected on the basis of minimum entropy of stumps. Stump is nothing but a base learner which is a tree with just one node and two leaf nodes, or it may be having many numbers of leaf nodes. Later than training set $(f_1, f_2, f_3, \cdots\cdots\cdots f_{n-1}, f_n)$ where $f_n \in \{0, 1\}$ is labeled class with weight of each record of size

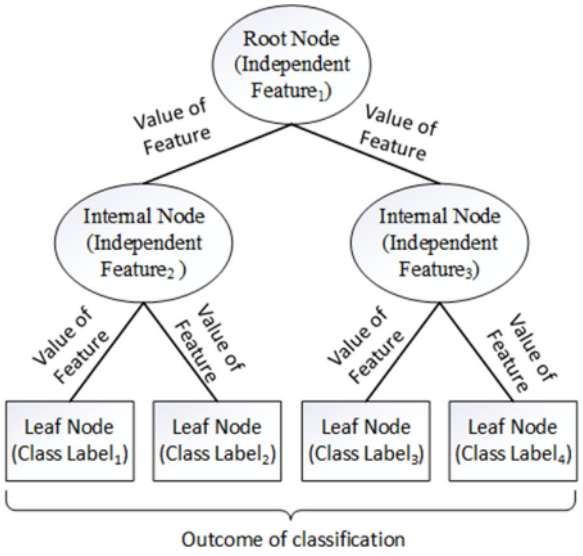

Fig. 4. Hierarchical structure of a decision tree.

k is calculated by $\frac{1}{k}$ are passed to the $model_1$ for training. If some records are misclassified by the $model_1$ then new model such as $model_2$ is created sequentially and only misclassified records with updated weights are passed to the $model_2$ to train the model, by doing this there is less chance of overfitting [41]. Simultaneously, if $model_2$ produces some more misclassifications, then errors and updated weights are passed to the next base learner such as $model_3$ for training. This is repetitive unless and until with specific number of base learners to minimize the errors in each cycle during training process. Finally, the test data will be fed into every model in the sequence after that weight of each model is combined using weighted vote to make an ultimate prediction.

(c) **Random Forest (RF)**: RF is a kind of ensemble learning classifier based on the bagging technique, which can be defined as an ensemble of decision trees in a randomized way [14] to make a forest. Fig. 6 shows the implementation of RF in which it works parallelly into two phases of randomization at each iteration; first phase of randomization is to create bootstrap dataset from the training set $\left(f_i^1, f_i^2, \cdots \cdots f_i^{n-1}, f_i^n\right); \forall i = 1, 2, \cdots \cdots \cdots \cdots k$ where $f^1, f^2, \cdots \cdots, f^{n-1}$ are independent features, $f^n \in \{0, 1\}$ is a dependent feature and k is total number of records). Bootstrap dataset is randomly picked any of the records such as $sample_1\left(f_j^1, f_j^2, \cdots \cdots f_j^{n-1}, f_j^n\right); \forall j = 2, 3, 5 \cdots \cdots \cdots \cdots$ from the training set. Records which are not present in the $sample_1$ is called out of bag records for $sample_1$ and further it is used to validate RF model.

In the second phase of randomization, to train every decision tree in the forest, a random decision tree is generated using a bootstrap dataset. In addition, the root node of the decision tree is decided randomly by the subset of features of $sample_1$at each step. Now, these subsets are candidates to becoming a root node by doing this it can classify the sample in a better way. This process is iterative to generate each DT in the forest. Finally, every DT in the forest produces a prediction for a single vote then by counting the majority vote from all the DTs prediction, results of the ensemble model is determined. A significant benefit of RF model is that it is more accurate than DT and AdaBoost on unseen data and less likely to become overfit [14,42]. Moreover, it requires minimal computational resources and does not require tree pruning, which is a time-consuming task [42].

Table 2
Features with their index and name.

Feature index	Feature name	Feature index	Feature name	Feature index	Feature name
F0	dur*$	F15	Sintpkt*@	F30	ct_srv_src*^
F1	proto**#	F16	dintpkt*@	F31	ct_state_ttl*+
F2	service**$	F17	sjit*@	F32	ct_dst_ltm*^
F3	state**$	F18	djit*@	F33	ct_src_dport_ltm*^
F4	spkts*$	F19	swin*%	F34	ct_dst_sport_ltm*^
F5	dpkts*$	F20	stcpb*%	F35	ct_dst_src_ltm*
F6	sbytes*$	F21	dtcpb*%	F36	is_ftp_login*+
F7	dbytes*$	F22	dwin*%	F37	ct_ftp_cmd*+
F8	rate*$	F23	tcprtt*@	F38	ct_flw_http_mthd*+
F9	sttl*$	F24	synack*@	F39	ct_src_ ltm*^
F10	dttl*$	F25	ackdat*@	F40	ct_srv_dst*^
F11	sload*$	F26	smeansz*%	F41	is_sm_ips_ports*+
F12	dload*$	F27	dmeansz*%	F42	attack_cat*~
F13	sloss*$	F28	trans_depth*%		
F14	dloss*$	F29	res_bdy_len*%		

Terms and Abbreviations: *Numeric type, **Categorical type, ~Labeled feature, #Flow features, $Basic features, %Content features, @Time features, +Additional generated features, ^Connection features.

Table 3
Distribution of samples of UNSW_NB15 training and testing dataset.

S.N.	Class types	Total samples in training set	Total samples in testing set	Total samples
1	DoS	12,264	4,089	16,353
2	Exploits	33,393	11,132	44,525
3	Fuzzers	18,184	6,062	24,246
4	Generic	40,000	18,871	58,871
5	Reconnaissance	10,491	3,496	13,987
6	Analysis	2,000	677	2,677
7	Backdoor	1,746	583	2,329
8	Shellcode	1,133	378	1,511
9	Worms	130	44	174
10	Normal	56,000	37,000	93,000
	Total	**1,75,341**	**82,332**	**2,57,673**

5. Result analysis and discussion

In this section, the results of the proposed intelligent cyber threats detection framework are discussed. This section comprises of four parts, in Section 5.1 description of dataset is described. Section 5.2 describes the experimental setup. In Section 5.3, results of the proposed models are discussed. Section 5.4 compares six state-of-the-art models with the proposed models in the same experimental situation.

5.1. Description of benchmark dataset UNSW-NB15

UNSW-NB15 is a network traffic-based dataset with modern low-footprint cyber threats, as suggested by Moustafa et al. [43] for cyber threats detection. IXIA Perfect Strom tool is used by the Australian Center for Cyber Security to develop the UNSW-NB15 dataset. It consists of real-world examples of threats and synthetic threats simulated using Pump, which can generate up to 100 GB of cap files each to simulate 9 different types of cyber threats: DoS, Exploits, Fizzers, Generic, Reconnaissance, Analysis, Backdoor, Shellcode, Worms, as shown in Table 3. Original dataset having 49 features belonging to the target feature with approximately 25,40,044 samples. Apart from this, openly accessible training and testing dataset are separated into 1,75,341 and 82,332 data instances respectively [43]. It is observed that, the training and testing dataset having 43 features with a labeled class as shown in Table 2 and majority of investigators have used this separated dataset for the experimental purpose. Therefore, for this experiment separated training and testing dataset is considered.

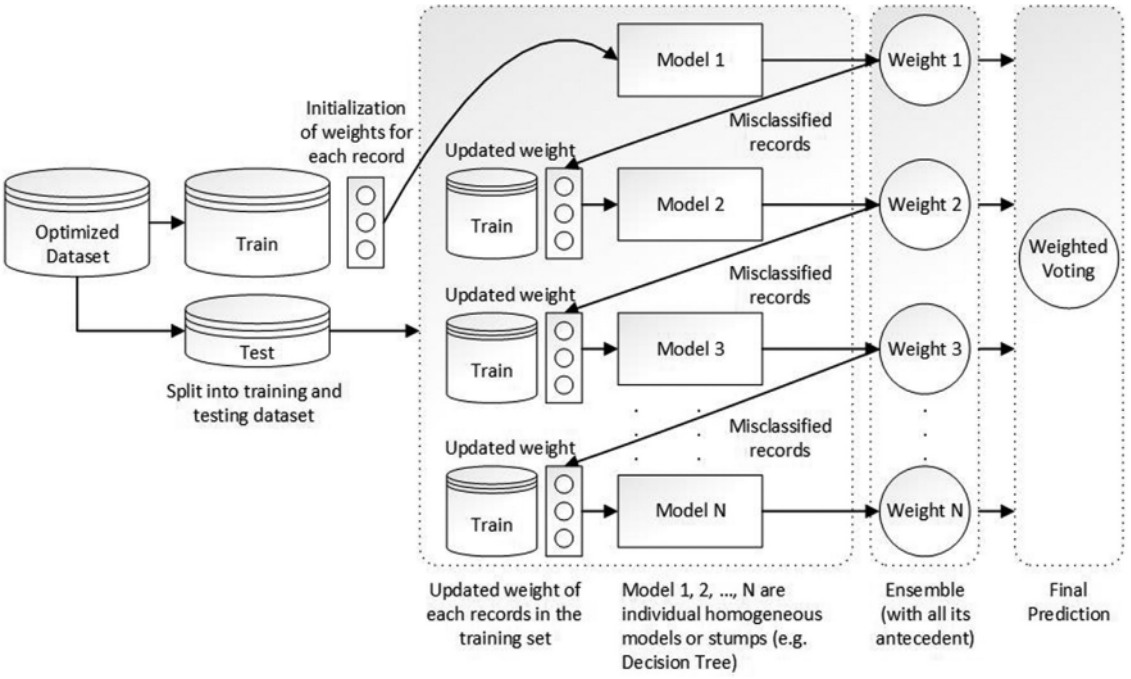

Fig. 5. Architecture of AdaBoost classifier for cyber threat detection.

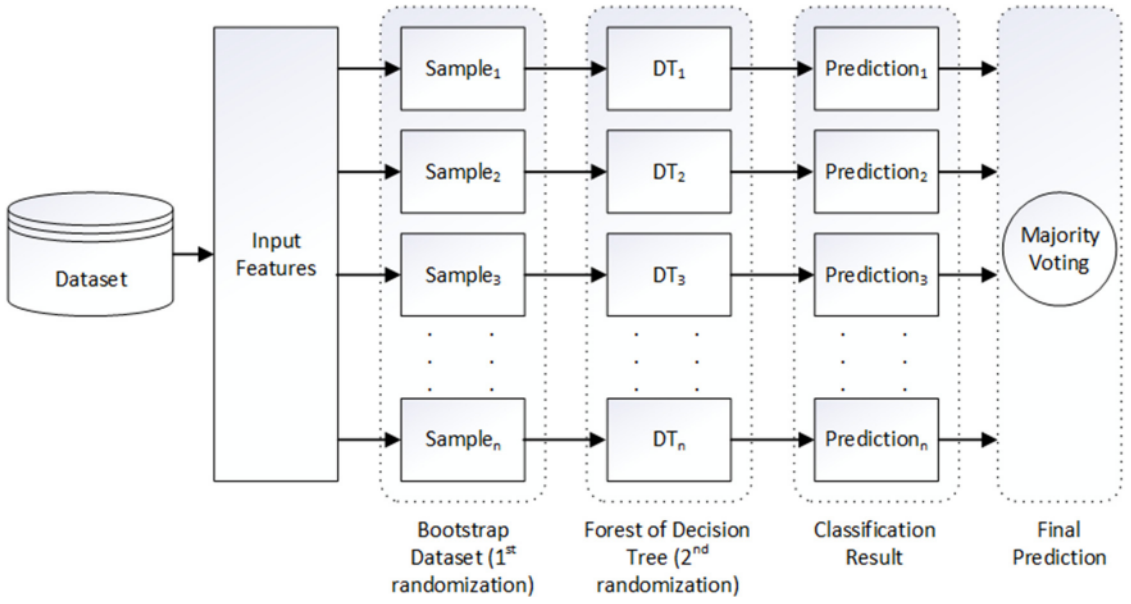

Fig. 6. Architecture of Random Forest classifier for cyber threat detection.

5.2. Experimental setup

This section describes the experimental environment, implementation details, and evaluation metrics for experimental evaluation:

(a) ***Experimental Environment***: Python 3.9 – SVC package is used for implementation of the proposed model. Experiments are carried out on Windows 10 PC. The experiments are performed on Spyder IDE using 2.20 GHz 8-Core Intel i7 CPU with 16 GB RAM.

(b) ***Implementation Details***: To find an optimized subset of features using BGSA and BGWO, different parameters are considered for the implementation, as described in Table 4. Further, to detect cyber threats, DT, AdaBoost, and RF are used to learn the model. In this work, hyperparameters are tuned for AdaBoost and RF from scikit-learn library in Python, as described in Tables 5 and 6.

(c) ***Evaluation Metrics***: To measure the detection performance of the proposed models, four basic classification metrics i.e., number of True Positive, False Positive, True Negative, and False Negative (FN) are calculated, as described in the Table 7. Based on the above metrics, five evaluation metrics: Accuracy (ACC), F1-Score, False Positive Rate (FPR), Detection Rate (DR), Precision (PR) are calculated in this experiment to check the performance of the presented model. Where ACC measures the proportion of correctly classified network traffic [18]. In DR, actual attacks are compared to the percentage of correct predictions, which reflects an analysis model's ability to predict

Table 4
Experimental setup details for BGSA and BGWO.

Parameter	Description
Total count of independent features	42
Total count of dependent features	1 (Binary class)
Number of objects/epochs	10
Train-test split	68.05% for training and 39.95% for testing, as shown in Table 2.
Fitness function	Formulated and expressed in Eq. (23)
Number of iterations	BGSA: 180, BGWO:140
Population initialization scheme	rand ()
Activation function	Sigmoid
Classifier for evaluation	Decision Tree for both BGSA and BGWO

Table 5
Hyperparameter tuning for AdaBoost.

base_estimator	Decision Tree
n_estimator	500
learning_rate	0.1
algorithm	SAMME.R
random_state	None

Table 6
Hyperparameter tuning for RF.

max_depth	4
max_features	4
min_samples_leaf	1
min_samples_split	2
n_estimators	10

Table 7
Confusion matrix for binary class cyber threats detection.

		Actual class	
		Normal network traffic (Negative)	Malicious network traffic (Positive)
Predictive class	Normal network traffic (Negative)	Correct decision: TN	Type II Error: FN
	Malicious network traffic (Positive)	Type I Error: FP	Correct decision: TP

Table 8
Statistics of selected features by proposed feature selection techniques for the UNSW-NB15 dataset.

Technique	FS Techniques	Number of selected features	Index of selected subset of features
BGSA	GS-DT	8	$\{F_0, F_4, F_5, F_{10}, F_{27}, F_{30}, F_{31}, F_{38}\}$
BGWO	GW-DT	12	$\{F_0, F_2, F_3, F_4, F_6, F_{10}, F_{14}, F_{19}, F_{22}, F_{30}, F_{34}, F_{35}\}$
Proposed	GSGW-DT	4	$\{F_0, F_4, F_{10}, F_{30}\}$

Table 9
Cyber threats detection performance of the proposed model.

Proposed threat Detection models	Detection technique	ACC (%)	DR (%)	PR (%)	F1-Score (%)	FPR (%)
GSGW-DT-DT	Decision Tree	99.02	98.32	99.50	99.04	0.24
GSGW-DT-AB	AdaBoost	99.36	98.47	**99.94**	99.27	0.04
GSGW-DT-RF	Random Forest	**99.41**	**99.09**	99.92	**99.33**	**0.03**

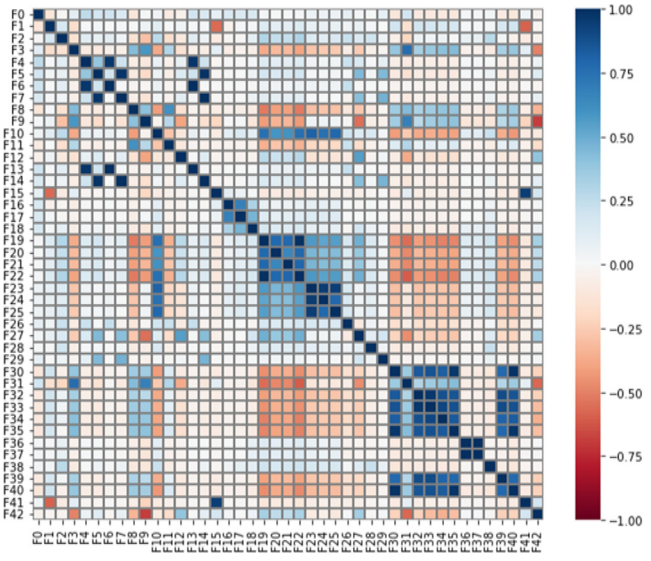

Fig. 7. Correlation coefficient matrix for all features.

malicious behavior [18]. A PR specifies the significance of the model by measuring the proportion of predicted attacks that actually occurred [18]. F1-Score estimates the harmonic mean of PR and DR. A high F1 score signals that threats are correctly identified. FPR evaluate the rate of normal instances that are identified as attack and overall count of normal instances [18].

5.3. Performance of the proposed work

In this section, the performance of the proposed FS techniques such as *GS-DT*, *GW-DT*, and *GSGW-DT* are evaluated using the UNSW-NB15 dataset. UNSW-NB15 includes all 9 types of threats, but attacks are not equally distributed, as shown in Table 3. Therefore, during the evaluation, repeated stratified-10-fold cross-validation approach is used to ensure that the classifier could generalize to unseen data. Table 8 shows the number of selected features obtained by the proposed FS techniques. Moreover, many researchers stated that a general improvement in accuracy is also associated with a higher FPR [14,28]. Therefore, the proposed model is designed to maintain the highest accuracy, detection rate, precision, and f1_score with lowest FPR. It can be observed form Table 9 data that the proposed model *GSGW-DT-RF* reaches the highest level of accuracy with 99.41% and the lowest FPR with 0.03%. However, the proposed model *GSGW-DT-AB* achieved a better result in terms of precision with 99.94% as compared to the proposed model such as *GSGW-DT_RF* and *GSGW-DT-DT*. In summary, based on the result described in Tables 8 and 9, it can be concluded that *GSGW-DT-RF* is a better model than other proposed models.

5.4. Validation of feature selection framework

In order to validate the number of selected features, Pearson Correlation Coefficient technique is applied to calculate the correlation coefficient scores among individual feature exist in the dataset. Figs. 7 and 8 shows the correlation coefficient matrix for all feature versus selected features by the proposed method named *GSGW-DT*. Fig. 8 shows that all the selected independent features such as F_0, F_4, F_{10}, and F_{30} have correlation coefficient values less than 0.50, which implies that there is low correlation among independent features. Due to this, the proposed model has achieved the best classification results.

5.5. Comparison of the proposed framework with existing techniques

This section compares proposed intelligent cyber threat detection models with six state-of-the-art models using the same experimental scenario. Table 10 shows the comparison results in terms of the number of selected features, accuracy, detection rate, precision, F1-score, and FPR. The proposed model *GSGW-DT-RF* records the highest accuracy with 99.41%, detection rate with 99.09%, F1-Score with 99.33%, and

Table 10
Comparison of proposed cyber threats detection framework with other metaheuristic-based compelling models in the same experimental scenarios.

References	Model name	Selected features	Detection technique	ACC (%)	DR (%)	PR (%)	F1-Score (%)	FPR (%)
Vajiheh et al. [11]	ABC-AFS	5	CART	98.60	99.0	98.70	98.88	0.13
Nazir et al. [14]	TS-RF	16	RF	83.12	86.22	86.32	86.21	3.7
Alazzam et al. [18]	Cosine-PIO	5	DT	91.70	89.40	90.93	90.90	0.34
Gauthama et al. [20]	HC-IBGSA SVM	30	SVM	94.11	98.47	98.74	98.62	2.18
Kumar et al. [21]	Rule-based IDS	13	DT	84.83	90.32	57.01	68.13	2.01
Tama et al. [24]	TES-IDS	19	TES	91.27	91.30	91.60	91.50	8.90
Proposed model	GSGW-DT-DT	4	DT	99.02	98.32	99.50	99.04	0.24
Proposed model	GSGW-DT-AB	4	AB	99.36	98.47	99.94	99.27	0.04
Proposed model	GSGW-DT-RF	4	RF	99.41	99.09	99.92	99.33	0.03

Terms & Abbreviations: TS-RF: Tabu Search - Random Forest; ABC-AFS: Artificial Bee Colony- Artificial Fish Swarm; Cosine-PIO: Cosine Pigeon Inspired Optimization; HC-IBGSA SVM: Hyper Clique—Improved Binary Gravitational Search Algorithm based Support Vector Machine; TES-IDS: Two stage classifier-based IDS; RF: Random Forest; CART: Classification and Regression Tree; DT: Decision Tree; SVM: Support Vector Machine; TES: Two stage ensemble; AB: AdaBoost.

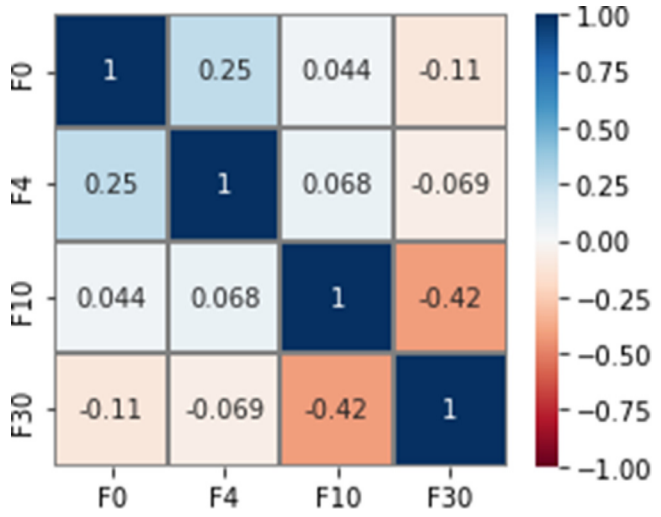

Fig. 8. Correlation coefficient matrix for selected features.

the lowest FPR with 0.03% with minimum number of features (4 features).

Fig. 9 illustrates the result analysis of the proposed models such as *GSGW-DT-DT*, *GSGW-DT-AB*, *GSGW-DT-RF* in terms of accuracy and, also compare its performance with the six state-of-the-art models. It is observed from Fig. 7 that proposed *GSGW-DT-RF* model perform better than the remaining models with 99.41% accuracy. Moreover, two other proposed models, such as *GSGW-DT-AB* and *GSGW-DT-DT*, achieved the second and third highest accuracy with 99.36% and 99.02%, respectively among other existing models. This is due to more robust optimized features provided by the proposed models as compared with the existing models. Moreover, an ensemble learning approach produces more accurate results than a single classifier.

Fig. 10 illustrates the result analysis of the proposed models such as *GSGW-DT-DT*, *GSGW-DT-AB*, *GSGW-DT-RF* in terms of detection rate and, also compare its performance with the six state-of-the-art models. It is observed from Fig. 10 that proposed *GSGW-DT-RF* model perform better than the remaining models with 99.09% detection rate. This is because of the fact that the proposed model obtains more robust, optimized features that minimize the problem of model overfitting and high detection errors. However, the model based on [11] have achieved the second-highest detection rate of 99% from other two proposed model such as *GSGW-DT-AB* and *GSGW-DT-DT*, but it is also significant to note that the number of selected features in *GSGW-DT-AB* and *GSGW-DT-DT* are 4, which is less than the selected features obtained by model [11].

Fig. 11 illustrates the result analysis of the proposed models such as *GSGW-DT-DT*, *GSGW-DT-AB*, *GSGW-DT-RF* in terms of precision rate and, also compare its performance with the six state-of-the-art models. It is observed from Fig. 11 that now proposed *GSGW-DT-AB* model perform better than the remaining models with 99.94% precision rate. However, in the evaluation of precision rate, two other proposed models such as *GSGW-DT-RF* and *GSGW-DT-DT*, achieved the second and third highest precision with 99.92% and 99.5%, respectively among other existing models. The reason behind this is that malicious network traffic should not be missed during the detection process. Therefore, the False Negative as defined in Table 7 should be as low as possible. In these circumstances, precision rate can be low, but detection rate should be high.

Fig. 12 illustrates the result analysis of the proposed models such as *GSGW-DT-DT*, *GSGW-DT-AB*, *GSGW-DT-RF* in terms of F1-Score and, also compare its achieving results with the six state-of-the-art models. As seen in Fig. 12, the proposed *GSGW-DT-RF* model performs better than the rest of the models, with 99.33% F1-score. Moreover, two other proposed models, such as *GSGW-DT-AB* and *GSGW-DT-DT*, achieved the second and third highest precision rates with 99.27% and 99.04%, respectively among other existing models. This is because F1-Score calculates the harmonic mean of precision and detection rate, and proposed models achieved the highest detection rate and precision as compared to existing models.

Fig. 13 illustrates the result analysis of the proposed models such as *GSGW-DT-DT*, *GSGW-DT-AB*, GSGW-*DT-RF* in terms of FPR and, also compare its performance with the six state-of-the-art models. It is observed from Fig. 13 that the proposed *GSGW-DT-RF* model performs better than the remaining models with the lowest FPR of 0.03%. This is due to ensemble learning techniques such as RF that combines multiple base classifiers to reduce false positive rate and produce more accurate solutions. Moreover, the proposed model *GSGW-DT-AB* has achieved the second lowest FPR with 0.04% among other models. This is due to ensemble learning techniques such as RF that combines multiple base classifiers to reduce false positive rate and produce more accurate solutions. However, model based on [11] have achieved the third lowest FPR with 0.13% from the proposed model *GSGW-DT-DT*, but it is also important to note that the number of selected features in *GSGW-DT-DT* is 4, which is less than the selected features observed by the model [11].

6. Conclusion

This paper proposes an intelligent cybersecurity threat detection framework in which the metaheuristic-based ensemble feature selection technique is used for the optimization of the features and ensemble learning-based classifiers are used to detect cyber threats. There are mainly three phases in the proposed framework. In the first phase, different data pre-processing steps are applied to IoT-based network traffic for data transformation and normalization. In the second phase, a novel feature selection approach named *GSGW-DT* is proposed that combines the subset of optimal features observed by *GS-DT* and *GW-DT* to get a single and more optimized subset of features. Then, a base classifier such as DT, and ensemble classifiers such as AdaBoost, and RF are employed separately in the third phase as analytical tools to

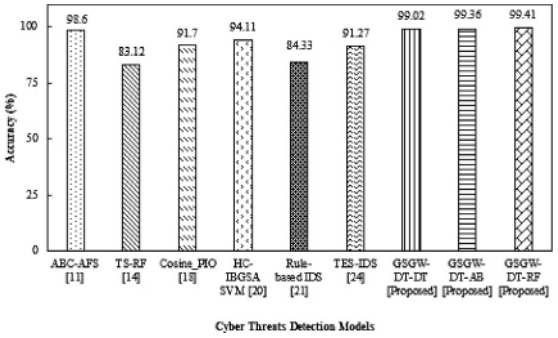

Fig. 9. Result evaluation of the proposed models in terms of accuracy.

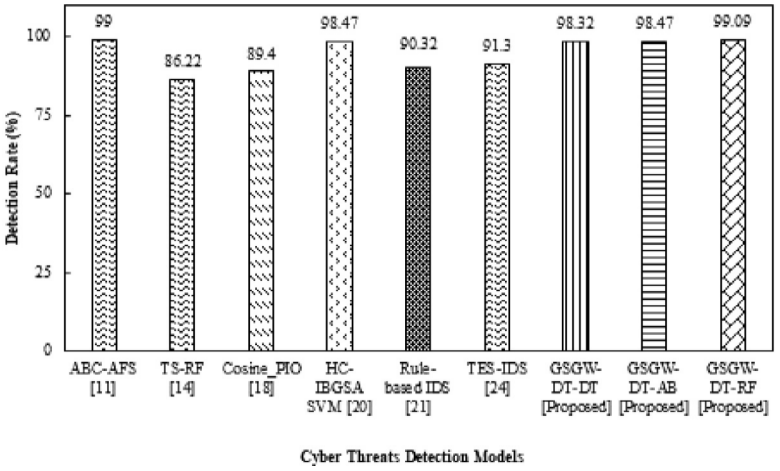

Fig. 10. Result evaluation of the proposed models in terms of detection rate.

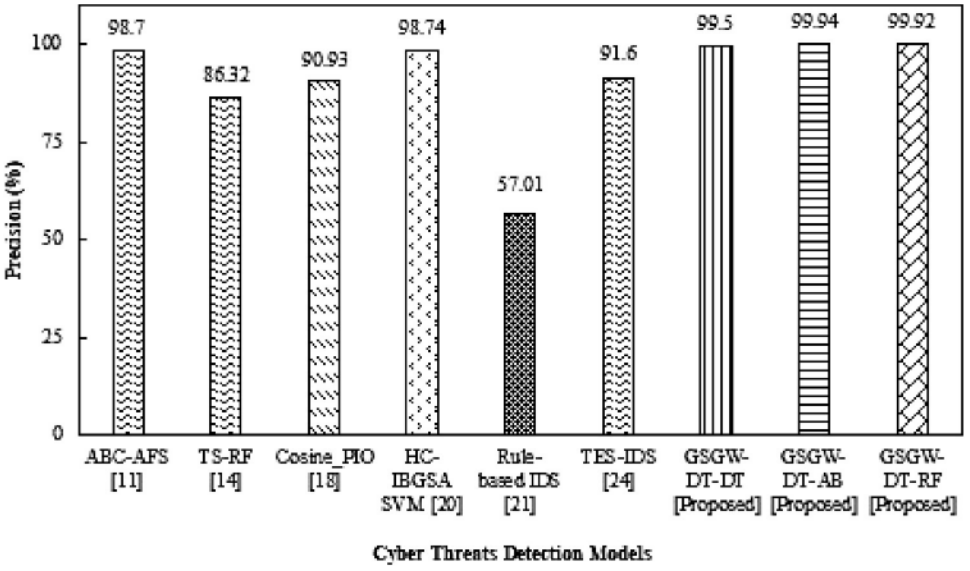

Fig. 11. Result evaluation of the proposed models in terms of precision rate.

enable fast and adequate decision-making in IoT-enabled networks. A benchmark dataset UNSW-NB15 is used to assess the performance of the proposed framework and compare it with recent existing solutions. Experimental results demonstrate that the proposed model *GSGW-DT-RF* achieved 99.41% accuracy, 99.04% DR, and 99.33% F1-score with the lowest FPR at 0.03% and 4 features out of 42 for the UNSW-NB15 dataset. However, another proposed model *GSGW-DT-AB* shows better results in terms of PR at 99.94% among the other models. In the future, the model capability will be enhanced to address zero-day cyber threats originating from IoT network traffic.

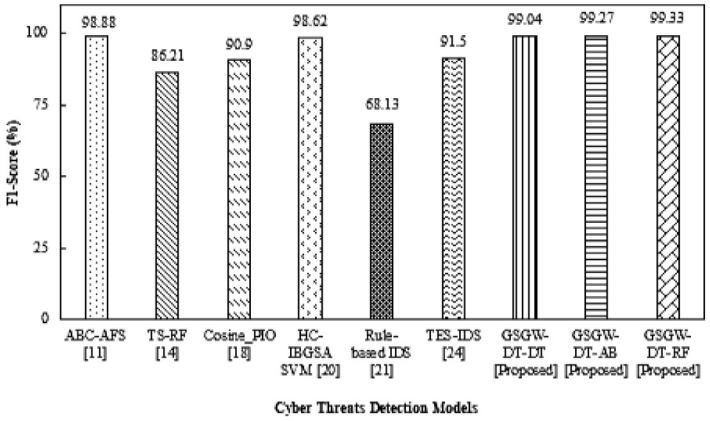

Fig. 12. Result evaluation of the proposed models in terms of F1-Score.

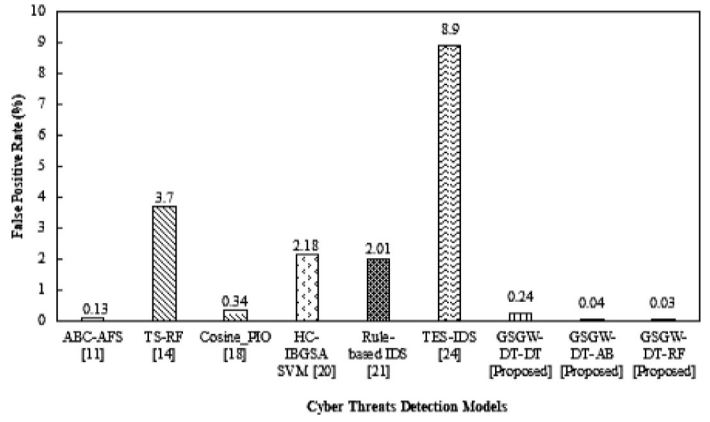

Fig. 13. Result evaluation of the proposed models in terms of FPR.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] O. Abu Alghanam, W. Almobaideen, M. Saadeh, O. Adwan, An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning, Expert Syst. Appl. 213 (PA) (2023) 118745, http://dx.doi.org/10.1016/j.eswa.2022.118745.

[2] P. Kumar, G.P. Gupta, R. Tripathi, Toward design of an intelligent cyber attack detection system using hybrid feature reduced approach for IoT networks, Arab. J. Sci. Eng. 46 (4) (2021) 3749–3778, http://dx.doi.org/10.1007/s13369-020-05181-3.

[3] R. Kumar, A. Aljuhani, P. Kumar, A. Kumar, A. Franklin, A. Jolfaei, Blockchain-enabled secure communication for unmanned aerial vehicle (UAV) networks, in: DroneCom 2022 - Proc. 5th Int. ACM Mobicom Work. Drone Assist. Wirel. Commun. 5G beyond, 2022, pp. 37–42, http://dx.doi.org/10.1145/3555661.3560861.

[4] Z. Chen, et al., Machine learning-enabled IoT security: Open issues and challenges under advanced persistent threats, ACM Comput. Surv. (2022) 1–35, http://dx.doi.org/10.1145/3530812.

[5] K.C. Darrell Etherington, Large DDoS attacks cause outages at Twitter, Spotify, and other sites, Techcrunch (2016) https://techcrunch.com/2016/10/21/many-sites-including-twitter-and-spotify-suffering-outage/#:~:text=Twitter%2CSoundCloud%2CSpotify%2C Shopify, first reported on Hacker News.

[6] Top 5 shocking IoT security breaches of 2019, PentaSecurity (2019) https://www.pentasecurity.com/blog/top-5-shocking-iot-security-breaches-2019/.

[7] Christine Buurma, Alyza Sebenius, Ransomware Shuts Gas Compressor for Days in Latest Attack, Bloomberg, 2020, https://www.bloomberg.com/news/articles/2020-02-18/ransomware-shuts-u-s-gas-compressor-for-2-days-in-latest-attack#xj4y7vzkg.

[8] Google, How Google Cloud Blocked Largest Layer 7 DDoS Attack Yet, 46 Million Rps | Google Cloud Blog, Google Cloud Blog, 2022, https://cloud.google.com/blog/products/identity-security/how-google-cloud-blocked-largest-layer-7-ddos-attack-at-46-million-rps.

[9] Statita Research Depratment, Internet of things- number of connected devices worldwide 2015–2025, Statista (2016) https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[10] M. Taneja, A. Davy, Resource aware placement of IoT application modules in fog-cloud computing paradigm, in: Proc. IM 2017-2017 IFIP/IEEE Int. Symp. Integr. Netw. Serv. Manag., 2017, pp. 1222–1228, http://dx.doi.org/10.23919/INM.2017.7987464.

[11] V. Hajisalem, S. Babaie, A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection, Comput. Netw. 136 (2018) 37–50, http://dx.doi.org/10.1016/j.comnet.2018.02.028.

[12] Y. Zhou, G. Cheng, S. Jiang, M. Dai, Building an efficient intrusion detection system based on feature selection and ensemble classifier, Comput. Netw. 174 (March) (2020) http://dx.doi.org/10.1016/j.comnet.2020.107247.

[13] D. Stern, N. Finkelstein, J. Stone, J. Latting, C. Dornsife, Statistics Published By Symantec Internet Security Threat Report 2019, vol. 24, Sch. To Work, 2020, pp. 12–15, http://dx.doi.org/10.4324/9780203046418-6.

[14] A. Nazir, R.A. Khan, A novel combinatorial optimization based feature selection method for network intrusion detection, Comput. Secur. 102 (2021) 102164, http://dx.doi.org/10.1016/j.cose.2020.102164.

[15] P. Kumar, R. Kumar, S. Garg, A Secure Data Dissemination Scheme for IoT-Based E-Health Systems using AI and Blockchain, GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 1397–1403.

[16] P. Kumar, G.P. Gupta, R. Tripathi, A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks,

J. Ambient Intell. Humaniz. Comput. 12 (10) (2021) 9555–9572, http://dx.doi.org/10.1007/s12652-020-02696-3.
[17] S. Garg, et al., En-ABC: An ensemble artificial bee colony based anomaly detection scheme for cloud environment, J. Parallel Distrib. Comput. 135 (2020) 219–233, http://dx.doi.org/10.1016/j.jpdc.2019.09.013.
[18] H. Alazzam, A. Sharieh, K.E. Sabri, A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer, Expert Syst. Appl. 148 (2020) http://dx.doi.org/10.1016/j.eswa.2020.113249.
[19] R. SaiSindhuTheja, G.K. Shyam, An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment, Appl. Soft Comput. 100 (2021) 106997, http://dx.doi.org/10.1016/j.asoc.2020.106997.
[20] M.R. Gauthama Raman, et al., An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm, 53, (5) Springer, Netherlands, 2020.
[21] V. Kumar, D. Sinha, A.K. Das, S.C. Pandey, R.T. Goswami, An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset, Clust. Comput. 23 (2) (2020) 1397–1418, http://dx.doi.org/10.1007/s10586-019-03008-x.
[22] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, S. Garg, M.M. Hassan, BDTwin: An integrated framework for enhancing security and privacy in cybertwin-driven automotive industrial internet of things, IEEE Int. Things J. 9 (18) (2022) 17110–17119, http://dx.doi.org/10.1109/JIOT.2021.3122021.
[23] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, S. Garg, M.M. Hassan, A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network, J. Parallel Distrib. Comput. 164 (2022) 55–68, http://dx.doi.org/10.1016/j.jpdc.2022.01.030.
[24] B.A. Tama, M. Comuzzi, K.H. Rhee, TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system, IEEE Access 7 (2019) 94497–94507, http://dx.doi.org/10.1109/ACCESS.2019.2928048.
[25] S. Dwivedi, M. Vardhan, S. Tripathi, Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection, Clust. Comput. 24 (3) (2021) 1881–1900, http://dx.doi.org/10.1007/s10586-020-03229-5.
[26] R. Kumar, P. Kumar, R. Tripathi, G.P. Gupta, A.K.M.N. Islam, M. Shorfuzzaman, Permissioned blockchain and deep learning for secure and efficient data sharing in industrial healthcare systems, IEEE Trans. Ind. Inform. 18 (11) (2022) 8065–8073, http://dx.doi.org/10.1109/TII.2022.3161631.
[27] P. Kumar, R. Kumar, G.P. Gupta, R. Tripathi, G. Srivastava, P2TIF: A blockchain and deep learning framework for privacy-preserved threat intelligence in industrial IoT, IEEE Trans. Ind. Inform. 18 (9) (2022) 6358–6367, http://dx.doi.org/10.1109/TII.2022.3142030.
[28] A. Oseni, et al., An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks, IEEE Trans. Intell. Transp. Syst. (2022) 1–15, http://dx.doi.org/10.1109/TITS.2022.3188671.
[29] P. Kumar, G.P. Gupta, R. Tripathi, S. Garg, M.M. Hassan, DLTIF: Deep learning-driven cyber threat intelligence modeling and identification framework in IoT-enabled maritime transportation systems, IEEE Trans. Intell. Transp. Syst. 24 (2) (2021) 1–10, http://dx.doi.org/10.1109/tits.2021.3122368.
[30] A.K. Dey, G.P. Gupta, S.P. Sahu, Hybrid meta-heuristic based feature selection mechanism for cyber-attack detection in IoT-enabled networks, Procedia Comput. Sci. 218 (2023) 318–327, http://dx.doi.org/10.1016/j.procs.2023.01.014.
[31] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Inf. Sci. (Ny). 179 (13) (2009) 2232–2248, http://dx.doi.org/10.1016/j.ins.2009.03.004.
[32] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, BGSA: Binary gravitational search algorithm, Nat. Comput. 9 (3) (2010) 727–745, http://dx.doi.org/10.1007/s11047-009-9175-3.
[33] M. Taradeh, et al., An evolutionary gravitational search-based feature selection, Inf. Sci. (Ny). 497 (2019) 219–239, http://dx.doi.org/10.1016/j.ins.2019.05.038.
[34] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.
[35] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381, http://dx.doi.org/10.1016/j.neucom.2015.06.083.
[36] J. Rojas-Delgado, R. Trujillo-Rasúa, R. Bello, A continuation approach for training Artificial Neural Networks with meta-heuristics, Pattern Recognit. Lett. 125 (2019) 373–380, http://dx.doi.org/10.1016/j.patrec.2019.05.017.
[37] Y. Wang, D. Wang, N. Geng, Y. Wang, Y. Yin, Y. Jin, Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection, Appl. Soft Comput. 77 (2019) 188–204, http://dx.doi.org/10.1016/j.asoc.2019.01.015.
[38] B. Selvakumar, K. Muneeswaran, Firefly algorithm based feature selection for network intrusion detection, Comput. Secur. 81 (2019) 148–155, http://dx.doi.org/10.1016/j.cose.2018.11.005.
[39] S. Peddabachigari, A. Abraham, J. Thomas, Intrusion detection systems using decision trees and support vector machines, Int. J. Appl. Sci. Comput. 11 (3) (2004) 118–134.
[40] Y. Zhou, T.A. Mazzuchi, S. Sarkani, M-AdaBoost-A based ensemble system for network intrusion detection, Expert Syst. Appl. 162 (August) (2020) 113864, http://dx.doi.org/10.1016/j.eswa.2020.113864.
[41] W. Hu, S. Member, W. Hu, S. Maybank, AdaBoost-based algorithm for network, IEEE Trans. Syst. Man. Cybern. 38 (2) (2008) 577–583.
[42] I. Ahmad, M. Basheri, M.J. Iqbal, A. Rahim, Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection, IEEE Access 6 (2018) 33789–33795, http://dx.doi.org/10.1109/ACCESS.2018.2841987.
[43] N. Moustafa, J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc. 2015, 2015, http://dx.doi.org/10.1109/MilCIS.2015.7348942.