International Journal of Information Security (2023) 22:1177–1195
https://doi.org/10.1007/s10207-023-00684-0

REGULAR CONTRIBUTION

A novel hybrid hunger games algorithm for intrusion detection systems based on nonlinear regression modeling

Shahriar Mohammadi[1] · **Mehdi Babagoli**[1]

Accepted: 10 March 2023 / Published online: 11 April 2023

Abstract
Along with the advancement of online platforms and significant growth in Internet usage, various threats and cyber-attacks have been emerging and become more complicated and perilous in a day-by-day base. Anomaly-based intrusion detection systems (AIDSs) are lucrative techniques for dealing with cybercrimes. As a relief, AIDS can be equipped with artificial intelligence techniques to validate traffic contents and tackle diverse illicit activities. A variety of methods have been proposed in the literature in recent years. Nevertheless, several important challenges like high false alarm rates, antiquated datasets, imbalanced data, insufficient preprocessing, lack of optimal feature subset, and low detection accuracy in different types of attacks have still remained to be solved. In order to alleviate these shortcomings, in this research a novel intrusion detection system that efficiently detects various types of attacks is proposed. In preprocessing, Smote-Tomek link algorithm is utilized to create balanced classes and produce a standard CICIDS dataset. The proposed system is based on gray wolf and Hunger Games Search (HGS) meta-heuristic algorithms to select feature subsets and detect different attacks such as distributed denial of services, Brute force, Infiltration, Botnet, and Port Scan. Also, to improve exploration and exploitation and boost the convergence speed, genetic algorithm operators are combined with standard algorithms. Using the proposed feature selection technique, more than 80 percent of irrelevant features are removed from the dataset. The behavior of the network is modeled using nonlinear quadratic regression and optimized utilizing the proposed hybrid HGS algorithm. The results show the superior performance of the hybrid algorithm of HGS compared to the baseline algorithms and the well-known research. As shown in the analogy, the proposed model obtained an average test accuracy rate of 99.17%, which has better performance than the baseline algorithm with 94.61% average accuracy.

Keywords Intrusion detection system · Hybrid meta-heuristic algorithm · Hunger games search · Grey wolf optimization · Nonlinear regression · SMOTE-Tomek

1 Introduction

Network and Internet-based technologies have penetrated into all personal, social, political, and cultural aspects of human life due to their benefits and facilities. According to Statista reports, about 5 billion (65%) of the world's population uses the Internet, while cyber-attacks in 2020 increased by about 50% compared to the past year. In the last 3 years, in addition to the technology development, the global spread of the COVID-19 pandemic has also accelerated the growth in Internet users, which has equalized the risks and benefits of the Internet. CERT[1] announced that, in 2025, the financial injuries caused by cyberattacks might reach 11 billion dollars per year, combined with the loss of the trust of internet users. Any intrusion or malicious activity on network vulnerabilities, computers, or information systems may compromise system security parameters such as confidentiality, integrity, and availability (CIA) [1]. The malicious activities of intruders in cyberspace are damaging computer networks with new and sophisticated techniques to steal information, gain illegal profits, discover new vulnerabilities, and disable services. Intrusion detection systems (IDSs) play an essential role in network security, with the first idea being proposed

✉ Shahriar Mohammadi
Mohammadi@kntu.ac.ir

Mehdi Babagoli
Mehdi.babagoli@email.kntu.ac.ir

[1] Industrial Engineering Department, KN Toosi University of Technology, Tehran, Iran

[1] Computer Emergency Response Team.

in 1980 [2]. In the first edition (signature based), the patterns of different attacks were stored in a database, and the system notified the network administrator of the occurrence of an attack by creating an alert if new traffic matched any of the patterns. Failure to detect unknown attacks due to the outmoded database is a critical disadvantage of this type of IDS, which leads to the vulnerability and inefficiency of the systems [3]. With the emergence of anomaly-based IDSs and machine learning techniques, the strength of these systems has increased significantly. Machine learning techniques and deep learning are sub-assemblies of artificial intelligence science, attracting the attention of many researchers in the field of network security in recent years [4]. Due to the dynamic changes in network behavior, artificial intelligence techniques can determine the pattern of legal and illicit traffics by analyzing network packets and then classify them as benign or attack traffic. Feature engineering is the main difference between machine learning and deep learning approaches [5]. Since machine learning algorithms do not perform well in large and unbalanced data, they need efficient preprocessing and feature selection methods to improve the algorithm's diagnosis performance by removing redundant features and data. On the other hand, deep learning methods employ complex strategies on different layers to perform feature selection operations. They can be used for raw and huge data, but require hardware with high processing power, which is expensive [6]. Many studies have been conducted on the intrusion detection based on machine learning to increase detection accuracy, reduce the false alarm rate, select the best subset, and generalize different attacks. On the contrary, although CICIDS[2] is a recently proposed real-world dataset, it suffers from the problem of unbalanced classes and data redundancy, which directly impacts the algorithm's performance [3]. Due to the dynamic nature of computer networks, modeling the behavior of the network statically is considered NP[3]-Hard problems that can be solved using effective optimization methods such as machine learning [7] and meta-heuristic algorithms. Meta-heuristic algorithms are potential solutions and one of the problem-independent optimization methods that are very effective and widely used to solve NP-Hard problems and complex optimization. Exploration and exploitation are two performance cornerstones of meta-heuristic algorithms that investigate search space to find the optimal solution [8]. Meta-heuristic algorithms are inspired from natural phenomena and divided into two categories based on single and population-based solutions. Algorithms based on a single solution boost the local search process based on a single solution and improve the solution in its neighborhood. In contrast, population-based algorithms guide the search process to the optimal solution by maintaining multiple solutions at different points of the search space [9]. Even though the performance of meta-heuristic algorithms is still a "black box" on why certain meta-heuristics perform better on specific optimization problems, a different research has proposed modified or hybrid versions of algorithms to improve the performance and robustness of the classical algorithms [10, 11]. In general, modification of the components of meta-heuristic algorithms and combining them are carried out to enhance the efficiency and increase the convergence speed and escapes from the local optimum.

Due to some shortcomings in this area in case of imbalanced and outdated data usage, curse of dimensionality, high false positive rate and low accuracy and uncertain network's behavior [12], the main objective of this study was to provide an accurate and robust IDS using a comprehensive dataset, effective preprocessing methods, high-performance meta-heuristic algorithms, and a sophisticated model. Initially, CICIDS dataset was selected to evaluate the model efficiency. In order to tackle the imbalanced class problem, SMOTE Tomek was utilized in the preprocess step. In the feature selection phase, the binary gray wolf algorithm (GWO) was used to search and discover the best feature subset. Finally, the behavior of the network was modeled with a strong nonlinear regression model and optimized using a new meta-heuristic algorithm called the hunger games search (HGS). The implemented meta-heuristic algorithms were combined with genetic operators to improve the performance and efficiency. In both feature selection and intrusion detection strategies, mutation and crossover operators related to the genetic algorithm (GA) were implemented by the mentioned algorithms. In order to clarify the process of this research, the general framework of the proposed IDS is depicted in Fig. 1, and the innovations of this research are summarized as follows.

- *Preprocessing* Utilizing the SMOTE Tomek method which consists of both under sampling and over sampling for resampling and balancing the CICIDS dataset classes.
- *Feature selection* Utilizing the hybrid GWO and GA (GWOGA) methods as a search method and random forest as a classifier of the wrapper algorithm.
- *Training and testing* Proposing a quadratic nonlinear regression model to simulate the behavior of the network and optimizing using high performance and novel hybrid HGS and GA (HGSGA) algorithms.
- *Evaluation* Fast convergence in training and detecting different types of attacks with high accuracy and low false alarm rates.

The following sections of this article are organized as follows: In Sect. 2, research related to this article is examined. In Sect. 3, concepts and methods used along with governing

[2] Canadian Institute for Cyber-security Intrusion Detection System.

[3] Non-deterministic polynomial-time.

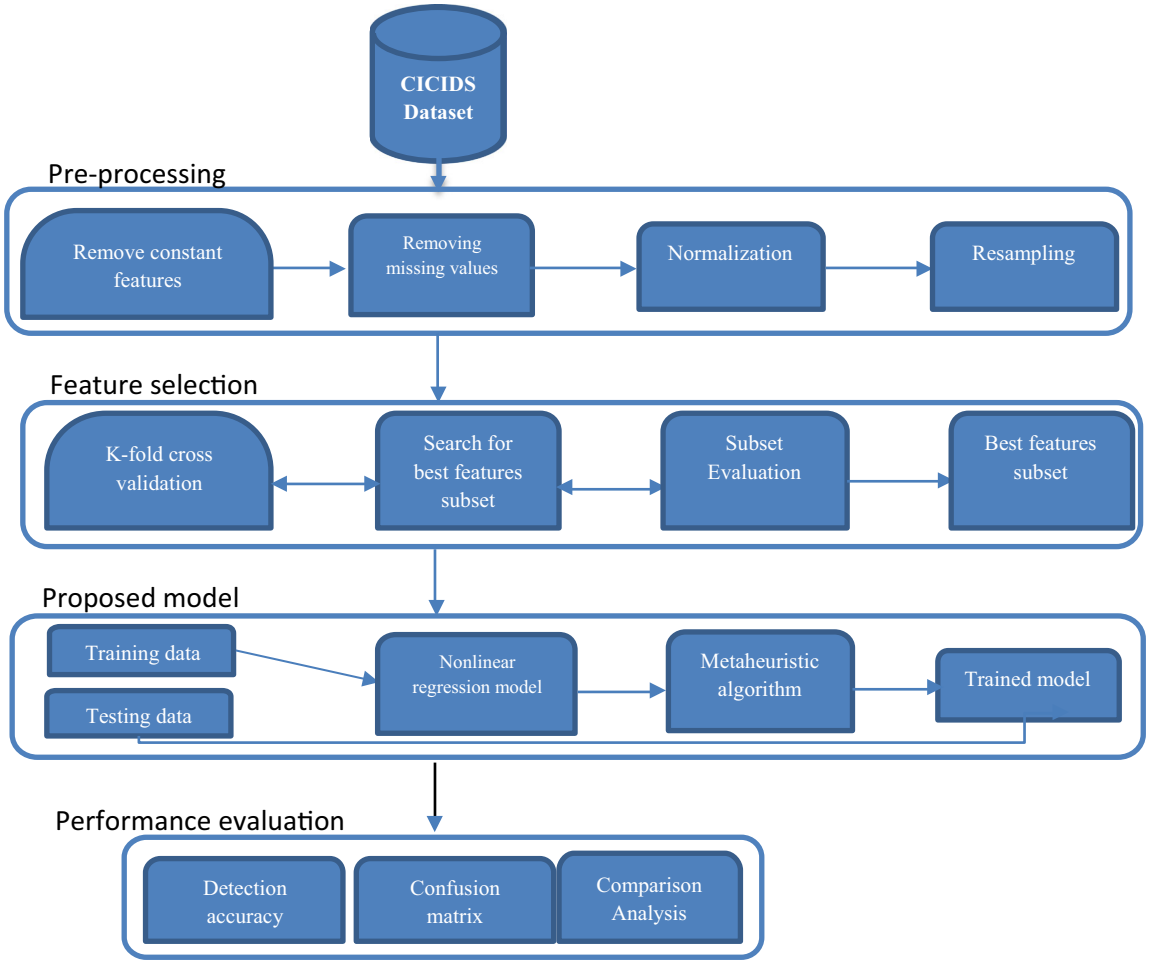

Fig. 1 Architecture of the proposed strategy for IDS

equations are described in detail. In Sect. 4, the results of the simulations are analyzed, and in Sect. 5, results and future suggestions of this research are presented.

2 Literature review

Nowadays, the rapid spread of Internet attacks besides the increasing number of internet users and the massive data on the network have brought the necessity of strengthening cyber security systems into more attention. Intrusion detection systems along with firewalls play a significant role in providing network security. Malicious traffic and connections can be blocked by firewalls using predefined provisions, but information is not accessible at the network layer. The IDS must monitor network traffic to detect intrusions and alert network administrators [13]. As shown in Table 1, intrusion detection systems are categorized as anomaly-based and signature-based [14]. Signature-based systems use information collected in the intrusion process to network and establish new patterns to make analogies with existing patterns, which lead to detecting intrusions. They are efficient

Table 1 Taxonomy of IDSs

Intrusion detection system taxonomy	Response method	Active
		Passive
	Source of data	Host-based
		Network-based
		Application-based
		Network behavior-based
		Wireless-based
		Mixed-based
	Detection method	Signature-based
		Anomaly-based
		Hybrid IDS
	Architecture	Centralized
		Distributed

in detecting known attacks for which knowledge is already available, but are still incapable of identifying unknown or novel attacks that do not exist on the knowledge base. On the other hand, any illegal action or significant behavior differing from authorized profile has been recognized by anomaly-based systems as an attack [15]. There are different categories of intrusion detection systems in the literature, the most important of which are shown in Table 1. These categories have been used in combinations in various studies. In this section, a detailed review of anomaly-based intrusion detection systems utilizing machine learning techniques and meta-heuristic algorithms, and their strategies, results and weaknesses is presented.

In [16], a distributed denial-of-service (DDoS) attack detection method in software-defined network (SDN) environments employing the meta-heuristic lion algorithm aiming at selecting the best feature subset was presented, in which the meta-heuristic lion algorithm was introduced. To classify authorized and fake traffics, convolutional neural network (CNN) algorithm was applied after selecting the best feature subsets. NSL-KDD[4] dataset was employed to train and test the proposed model. Prior to entering the preprocess phase in the proposed intrusion detection model, using the feature transformation method, all columns were converted to numerical values. Authorized and attack traffic columns were labeled by the values of 0 and 1, respectively. In order to evaluate the performance of the proposed model, Bee and Ant colony optimization algorithms were implemented along with CNN. The results proved higher functionality of the suggested intrusion detection model with DDoS attack detection accuracy of 98.2%, indicating 1% and 3% improvement, compared to Bee and Ant colony optimization algorithms.

In all networks, complete security has been achieved through a combination of firewalls, intrusion prevention equipment, and intrusion detection systems. Using an evolutionary algorithm and probabilistic dependency tree, Adjani et al. [17] have improved network intrusion detection by identifying effective features. In this research, NSL-KDD and VirusTotal datasets with different state-of-art machine learning classifiers have been used to train a dependency tree model. Each dataset was trained with a set of training data and then evaluated with experimental datasets. Each class of support vector machines on the datasets has a classification method, separately. To evaluate the feature subset, the average performance of the classification method was considered as a criterion. Using a binary string (0 or 1), the subsets were encoded. The features in this data set are divided into numerical and textual data into three categories: basic, content, and traffic. Basic features cause delays in the intrusion detection process. Content Features have been used for identifying external-to-internal and user-to-root attacks. Traffic characteristics are features that determine the percentage of past connections to current connections with the same service and host and are called machine-based. For huge datasets, feature selection reduces the detection cost and time while increasing classifier efficiency. RF classifiers demonstrate better detection accuracy, and other classifiers provide the best detection accuracy with consistency measures for most of the attack classes. It is worth mentioning that for tree-based decision-making methods, when the nominal property has a large number of unique values, the learning time will be very long and learning will be slow. With the proposed method, the learning speed greatly increased and the accuracy was acceptable.

Pandey et al. [18] developed a meta-heuristic autoencoder deep learning-based model for the intrusion detection system. A multichannel autoencoder deep learning approach was developed to determine the accuracy and false alarm rate of the intrusion detection systems. Initially, two different autoencoders were trained using average and attack traffics, where feature dependencies were monitored. Then, to better differentiate between benign and illicit traffic, CNN learned the probable relationships between existing channels. The experimental results demonstrated significant progress, such as a decrease in the number of features, easy integration with neural networks, cutting down training time, and intrusion detection accuracy enhancement. In this study, to optimize the one-dimensional CNN model's topology, a genetic algorithm was applied in order to improve intrusion detection performance. In [19], an algorithm based on double particle swarm optimization (PSO) was developed to detect network intrusion. Since irrelevant and redundant features cause high false alarm rate and low intrusion detection accuracy, a double PSO-based algorithm was proposed to select the feature subset and hyperparameters of the model simultaneously. Two well-known datasets, NSL-KDD and CICIDS2017, were utilized in the experiments. Only 10% of CICIDS2017 datasets were selected randomly without replacement. The intensive quantitative analysis, Friedman test, and ranking procedures were carried out to evaluate the model's performance. The results indicated that the proposed algorithm resulted in significant improvement in the network intrusion detection by 4 to 6% and decreased the false alarm rate by 5–1% from the same models' corresponding values without pretrains in similar datasets. In [20], a new method has been introduced for enhancing intrusion detection performance. With increasing network attacks and intrusions in computer networks, the importance of developing security policies, documenting existing threats, and preventing individuals from violating security policies to secure information was identified. To evaluate model efficacy, the UNSW-NB15 benchmark dataset has been used in this research, which comprises 9 attack types, 49 class-labeled features, and 2,540,044 observations. The authors

[4] Network Security Laboratory-Knowledge Discovery in Databases.

intend to improve the accuracy and speed of the intrusion detection method (system). They have utilized random forest and PSO algorithms. As shown in the result, PSO-RF was focused on the applicability of the new cosmology. It was found that comparing the PSO-Xgboost model, PSO-RF method, and multi-objective particle swarm optimization approach, the proposed model reached 96.5% efficiency. Alzubi et al. [21] extracted significant features in the intrusion detection system by hybridization of metaheuristic GWO and PSO algorithms, employing NSL-KDD and UNSW-NB15 datasets. Support vector machine (SVM) classification algorithm and decision tree were used to evaluate feature subsets. A modified version of GWO and PSO algorithms and a combination of GWO and GA were implemented separately to evaluate and compare the proposed algorithm. The meta-heuristic algorithm was applied for deep-learning algorithm efficiency elaboration and adjusting its parameters. The results indicated that the proposed and SVM algorithms could outperform other existing solutions, as the detection accuracy improved by approximately 0.3–12% and the detection rate by 2–12%. In addition, it reduced the false alarm rate and the number of features by 4–43% and approximately 31–75%, respectively. Finally, the proposed approach declined the processing time by approximately 14–22% compared to novel approaches. Zhou et al. [22] concentrated on high-dimensional network traffic and imbalance dataset classes and consequently presented a model based on a meta-heuristic feature selection algorithm and ML-based IDS. In the first phase, a heuristic algorithm called CFS-BA[5] [23] was applied to choose the best feature subset based on the correlation between features and Bat algorithm. In the next step, ensembled learning technique was used where C4.5 algorithm and random forest were combined. The best performance was achieved through the voting technique. Three datasets, NSL-KDD, AWID[6], and CICIDS, were used to train and evaluate the purposes. The experimental results were promising with an accuracy of 99.81% with a subset of 10 features for the NSL-KDD dataset, and the obtained results for AWID provided an accuracy of 99.52% with a subset containing 8 features. The effect of feature selection on enhancing the performance of meta-heuristic-based IDS in a cyber-physical environment was investigated by Quincozes et al. [24]. F1-Score criteria for meta-heuristic adapted greedy randomized adaptive search procedure (GRASP) were applied to improve intrusion detection accuracy through binary, multi-class, and expert classifiers. The proposed GRASP feature selection algorithm encompassed solution hybridization, fitness function selection, proximity structure, and RCL generation. Binary, multi-class and expert classifiers were applied to train classifying algorithms on the datasets (WSN-FS[7] and SWaT[8]) including benign and illicit traffics. All three training approaches were capable of good results through an appropriate subset selection. Among 51 available features on the SWaT dataset, GRASP algorithm selected a reduced subset containing 5 features and applied random tree to evaluate the dataset. According to the results, 96.97% F1-Score and 99.65% accuracy were reached using the proposed algorithm. It indicated the usability of GRASP for feature selection and its capability in providing good results. The highest average results for WSN-FS dataset were provided by IDS where blackhole attacks were detected with up to 99.83% F1-Score and 99.99% accuracy, whereas the average for all attacks was 93.32% F1-Score and 99.62% accuracy. Ajdani et al. [12] proposed a hybrid methodology called real-value particle swarm optimization (RPSO) to adjust the coefficients of the regression-based model for IDS. In order to evaluate the performance of the proposed model, VirusTotal dataset was utilized to forecast the behavior of the intrusion detection system. As shown in the result, the proposed model obtained 0.0234 and 1.845 for root-mean-square error (RMSE) and mean absolute percentage error (MAE), respectively. The experimental results proved that the proposed RPSO-logistic regression technique outperforms the standard regression and backpropagation (BP) neural network models. Ajdani et al. [25] in another research proposed an analytical framework that is suitable for high-scale data. The VirusTotal dataset has been applied including 2.5 M scans of 30 months (from 2012 to 2015), which represents a huge amount of data. For feature extraction and classification, destructive data detection, user management, and data behavior are organized in parallel in the pipeline framework architecture (which resulted in more speed) and the data have been trained in sub-periods. In the essence, destructive data content is dynamic. So, the authors applied a modified vector machine algorithm for promoting SVM. The proposed algorithm checks the changes of labels at any moment, and so takes into account the possibility of transforming to destructive data. Three factors are considered in this regard: time, scalability, and users' information. Cross-validation for running labels has resulted in the scalability factor increasing. The method has an accuracy of 97% that can be fairly accepted.

Otair et al. [26] presented an IDS system in WSNs based on improved PSO and meta-heuristic GWO algorithms. The NSL KDD dataset was used to verify the proposed technique's functionality. The classification process was done using k-means and SVM algorithms. Accuracy, intrusion detection, false alarm rates, the number of features, and execution time criteria were applied to measure the technique's performance. The results demonstrated that using K-means

[5] Correlation-Based Feature Selection Technique-Bat algorithm.

[6] Aegean Wi-Fi Intrusion Dataset.

[7] Wireless sensor network-feature selection.

[8] Secure Water Treatment.

Table 2 Comparison of the related work

Studies	Preprocess	Dataset	Feature selection	Accuracy (%)
[16]	Nominal to numeric	NSL-KDD	Lion optimization algorithm	CNN: 98.2
[18]	Yes, but not mentioned	NSL-KDD, CICIDS, UNSW-NS12	Auto-encoder	CNN + GA: 95C NN + GA: 98 CNN + GA: 92
[19]	Nominal to numeric, normalization	NSL-KDD, CICIDS	Double PSO	DBN: 99.79 DBN: 99.91
[21]	Mapping, Transformation normalization	NSL-KDD, UNSW-NS12	GWO-PSO	SVM: 94.74
[22]	Filtration, transformation, normalization	NSL-KDD, CICICDS	CFS-BA	Ensemble: 99.52 Ensemble: 99.89
[24]	Not mentioned	WSN-FS, SWaT	GRASP	Random Tree: 99.99 Random Tree: 99.65
[26]	Transformation, normalization	NSL-KDD	GWO-PSO	SVM: 98.97 K-Mean: 74.48
[27]	Encoding, normalization	NSL-KDD	MGWO	SVM: 96
[28]	Normalization, nominal to numeric	NSL-KDD	Cuckoo search	SVM: 95.88

or SVM algorithms, the proposed technique would enhance GWO algorithm by incorporating PSO. To achieve the main goal, a quantitative research method was applied, and the enhancement was reflected in the IDS protection level. The proposed technique was compared with PSO and GWO to measure the improvement, separately. The results indicated that the method could outperform PSO and GWO in terms of accuracy, the intrusion detection rate, and the number of features. The best result was achieved in the combined method using SVM algorithm. The yielded intrusion detection accuracy was about 98.97% in the dataset containing 20 features. The improved gray wolf algorithm was used in [27] to search and select the best subset of features in the NSL-KDD dataset. To evaluate the performance of the proposed model, SVM algorithm was used for intrusion detection, with 3, 5 and 7 wolves being used to find the best number of wolves. According to the results, the IDS with seven wolves could overwhelm the other proposed algorithms. The proposed algorithm aimed to increase the intrusion detection rate and accuracy and lower process time in WSN by reduced false alarm rates, and the features resulting from IDSs. By increasing the number of wolves and using the multi-objective function, the overall performance of IDS enhanced. In terms of detection rates, accuracy, the total number of selected features, false alarm rates, and execution time, GWOSVM-IDS outperformed both original GWO and PSO. The GWOSVM-IDS technique was also more successful in terms of accuracy, the detection rate, false alarm rate, number of selected features, and execution time than GWO and PSO techniques. Imran et al. [28] presented cuckoo search optimization (CSO) coupled with SVM for IDS. The research included modules, such as preprocessing, feature selection, and classification. To increase the attack detection accuracy, preprocessing was done using minimum–maximum standardization to remove lost values and filter plugin specifications from the NSL KDD dataset. In the next step, the feature selection process was implemented using CSO algorithm to select the optimal features of NSL KDD cup dataset. Higher accuracy of (95.88%), recall (80.58%), sensitivity (94.86%), specificity (95.88%) and precision (98.2629%) were achieved through the proposed CSO along with SVM algorithm. In Table 2, a comparison of the related works along with their research methods and conclusion is provided.

In most of the aforementioned research, the NSL-KDD dataset, which is an old dataset not generalizable to the modern network, was employed. On the other hand, these data were collected using simulation, which cannot be relied upon in the real environment, and as a result, it is better to use the latest and comprehensive datasets such as CICIDS. CICIDS dataset is a new and reliable dataset in which the data are collected in a real network environment. Based on McAfee's 2016 report, this dataset could collect common attacks, which included different types of DDoS, DoS, Infiltration, Heartbleed, and web attacks. Principally, this dataset has 2.8 million data along with 80 network traffic characteristics collected by engineering the traffic characteristics related to the intrusion detection system [29]. Class imbalance, as the most significant challenge in CICIDS dataset, has been addressed utilizing preprocess and sampling techniques. As indicated in the literature, various methods have been used for feature selection, among which GWO algorithm was chosen for feature selection in this research. It was due to easier implementation, less storage space and calculations, faster convergence and fewer parameters, as well as stable performance [30].

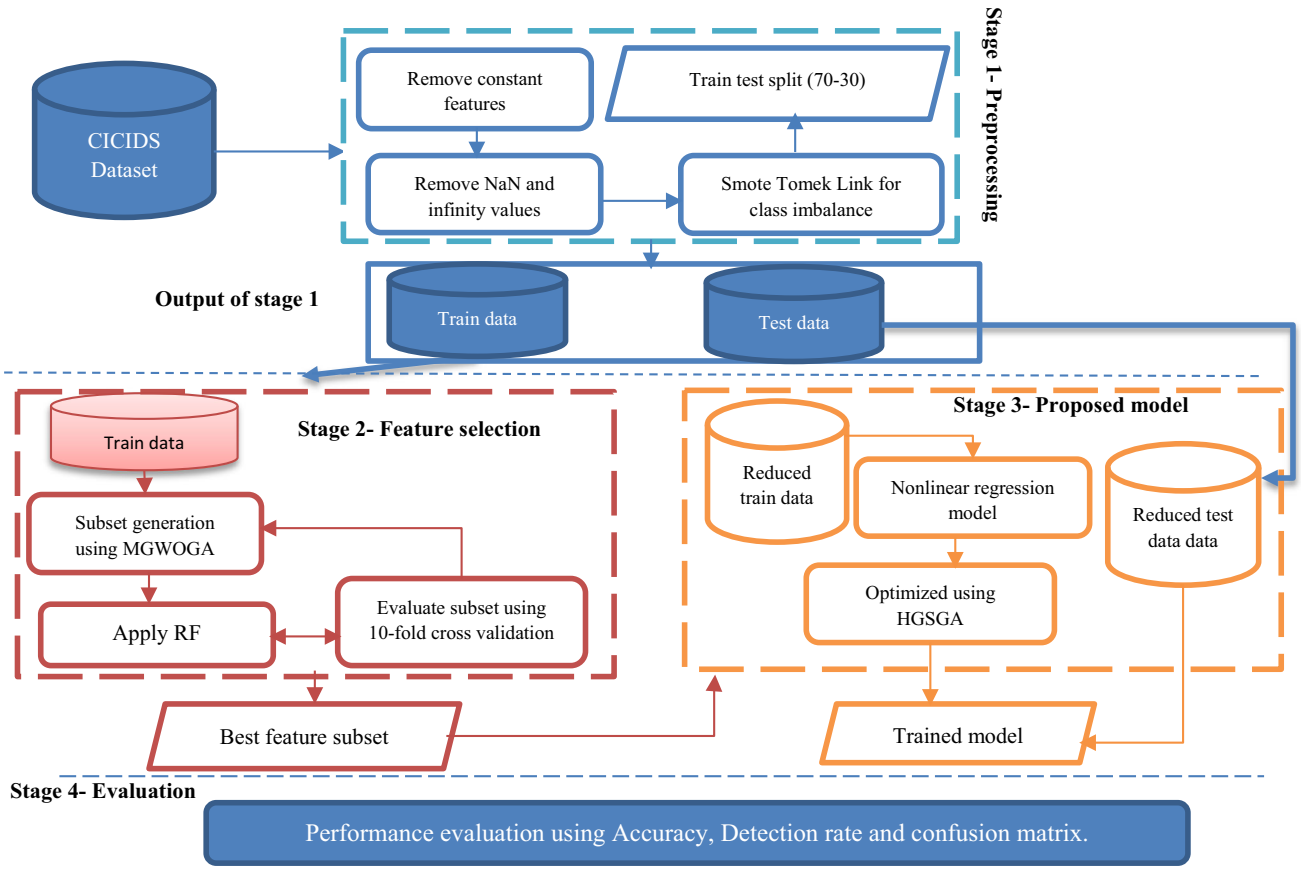

Fig. 2 Structure of the proposed framework

Also, the network behavior was converted to an optimization problem using nonlinear regression, and novel, and robust HS algorithm was considered for the optimization problem. To improve the performance of meta-heuristic algorithms, a combination of aforementioned algorithms was used.

3 The proposed method

In this section, the proposed intrusion detection system is described in detail to achieve the main goal of this research and solve the above-mentioned challenges. As shown in Fig. 2, the first stage was the selection of the dataset, which was used in this research from the CICIDS 2017 benchmark dataset. Then, preprocessing operations were applied to the data to clean, normalize and balance the dataset. After preparing the data, in the second stage, the wrapper method consisting of search and evaluation methods was used to remove irrelevant features and generate the best feature subset. In the next stage, the subset of features was embedded into the model and the intrusion detection operation was performed. Finally, the performance of the proposed mechanism was evaluated using different metrics. The figure below shows the detailed components of the proposed method in different parts separately. As shown in Fig. 2, the proposed method consisted of four main parts:

1. *Preprocessing* In this step, the features with constant values were removed from the dataset and the missing values were then eliminated in each row because of their small occurrence rate. In the next step, the min–max normalization was carried out on all columns, and finally, one of the new resampling methods was used to balance the classes of the dataset.
2. *Feature selection* In the feature selection stage, wrapper algorithm employed GWOGA to generate feature subsets and random forest (RF) algorithm to evaluate feature subsets.
3. *Proposed model* In order to consider intrusion detection, an optimization problem, quadratic nonlinear regression modeling was utilized and then the parameters of the model were optimized using hybrid meta-heuristic algorithms (HGSGA).
4. *Performance evaluation* The trained model in the previous section was evaluated using test and train accuracy and false alarm rates, and compared with classical algorithms and state-of-the-art intrusion detection models.

Table 3 CICIDS 2017 dataset description

	Value	Description
Total number of records	2,830,743	About 80% Benign traffic
Total number of features	79	**Frequency of each data type** Numerical: 60, Binary: 18, Class: 1
Type of attacks	Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS	**Number of instances** Brute Force: 13,835 DoS/DDoS: 294,506 Port Scan: 158,930 Infiltration: 36 Web attack: 2180

3.1 CICIDS dataset

Datasets tend to play an important role in performance evaluation of machine learning and meta-heuristic algorithms. As a result, in this research, a realistic, updated, and comprehensive dataset was selected to be utilized in the training phase. CICICDS dataset is a new benchmark dataset that consists of different types of attacks whose data are collected under real network scenarios. This dataset was collected in 5 days by the Canadian Institute for Cyber-security Intrusion Detection System, able to be used in two formats, PCAP and CSV [12, 25]. The major instances of the dataset were dedicated to DDoS attacks, with the lowest distribution to Heartbleed and SQL injection. Unbalanced distribution of legal traffic and attacks significantly affect the performance of machine learning algorithms and meta-heuristics that has not been considered in most of the recent related research. The details of this dataset are shown in Table 3.

As already mentioned, of the most important drawbacks of the utilized dataset is class imbalance and missing data, being solved in the preprocessing stage of this research. Due to the limited sample of some classes, Heartbleed, DoS, and DDoS attacks are categorized in one class, and FTP-Patator and SSH-Patator attacks are considered Brute Force class [31].

3.2 Preprocessing

The preliminary step in machine learning and meta-heuristic approaches is data cleaning and transforming datasets into a suitable format and performing data for another data processing procedure using preprocessing methods. Preprocessing is used to solve various problems of real-world data, such as incompleteness, inconsistency, trend existence, and error occurrence [32]. The CIC-IDS2017 dataset contains more than 2,800,000 samples, with a small number of missing and invalid values. The preprocessing steps were conducted as follows.

1. In the first preprocessing step, rows containing NaN and Infinity values were removed and then columns with fixed values were generally eliminated from the dataset. The constant features eliminated from the dataset were: *'Bwd PSH Flags', 'Fwd URG Flags', 'Bwd URG Flags', 'CWE Flag Count', 'Fwd Avg Bytes/Bulk', 'Fwd Avg Packets/Bulk', 'Fwd Avg Bulk Rate', 'Bwd Avg Bytes/Bulk', 'Bwd Avg Packets/Bulk', 'Bwd Avg Bulk Rate'*.
2. After redundant data elimination, in order to perform raw dataset usable for meta-heuristic algorithms and proposed mathematical model, nominal features were transformed into numerical data using the one-hot encoding method [26], after which data were rescaled using the min–max normalization method and mapped to [0,1] ranges. Normalization was the main step in preprocessing eliminating the discrimination effects of features on the model by bounding all features between the same range values. Equation 1 is related to the normalization process, where v' is a normalized value and v is the original value of the feature.

$$v' = \frac{v - \min(F)}{\max(F) - \min(F)} \tag{1}$$

3. In the last step, due to the dataset being prone to the issue of high-class imbalance, the resampling method was carried out to generate a standard and valid dataset. Resampling could also tackle common problems like large data sizes and hardware limitations. Principally, resampling is divided into oversampling and undersampling. In the first category, sampling is considered the minority class and duplicates or creates new synthetic instances to solve the class imbalance problem. In contrast, in oversampling, the instances from the majority class are deleted or merged. In the oversampling method, some random samples from the minority class are copied and no new information is added to the data. On the other hand, the under-sampling method removes some random samples from the majority class, which may discard potentially useful data [33]. One of the widely used oversampling methods is the SMOTE method, which utilizes the k-nearest neighborhood algorithm to find the nearest data of the random sample in the minority class and make synthetic data samples by combining the random data and k-nearest neighborhood sample [34]. Among the

several versions of SMOTE that have been represented in various studies, the SMOTE-Tomek link was used in this research in which two SMOTE and Tomek link algorithms were combined for oversampling and under-sampling, respectively. In the under-sampling approach, the instances with the lowest Euclidean distance in the majority class were removed from the dataset [35]. The pseudo-code is described below.

Algorithm 1. Pseudo code for SMOTE Tomek link

Input: Unbalanced dataset (D)
Output: Resampled dataset (D')

1. *Randomly select* x_i *in minority class*
2. *Identify K-nearest neighbor of* $x_i : \psi x_i$
3. ***Generate*** $x_{new} = x_i + (\hat{x}_i - x_i) \times \delta$
4. ***if*** *(! balancing ratio satisfied):*
   Go to 1
5. ***else****:*
   Remove Majority sample using Tomek
   TOMEK (l, m)

   {
   l is the nearest neighbor of m.
   m is the nearest neighbor of l.
   l and m belong to different classes.
   }
6. ***end if***
7. ***return D'***

3.3 Feature selection strategy

Feature selection is one of the most frequent and prominent techniques in data mining, and has become an indispensable component of the machine learning process. Feature selection algorithms are assessed to reduce the data volume while maintaining the quality of the data. This process is conducted by removing redundant and irrelevant features and obtaining an optimal feature subset. Therefore, reducing the size of the original dataset while maintaining performance accuracy is the primary goal of feature selection algorithms [35]. Feature selection problem is one of the most challenging tasks in machine learning, because a dataset with N features has 2N subsets. Since finding the best subset tends to be very expensive in high-dimensional datasets, various methods for feature selection have been proposed, many of which suffer from early convergence problems, high complexity, and high computational cost [36]. One of the most nominated methods in recent years has been meta-heuristic algorithms that have overcome the mentioned challenges. Principally, feature selection methods are divided into three categories: filter based, wrapper-based, and embedded. Meta-heuristic approaches are adopted to a subsection of wrapper algorithms (Fig. 3).

Meta-heuristic approaches are applied to wrapper algorithms to search and produce feature subsets by converting the features into a binary format. These subsets are evaluated using a classification method, and the best subset is selected [37]. In this research, the binary gray wolf meta-heuristic algorithm was used as a search method, and random forest (RF) algorithm was used to evaluate the subset of features to select the most relevant and salient feature subset. Gray wolf optimization (GWO) algorithm is a recently proposed swarm intelligence technique that has gained wide acceptance in the optimization community. Hunting behavior and dominance of the gray wolf in nature are imitated by this algorithm [38]. GWO has been applied to a large extent for various problems in different domains. Gray wolves live and hunt in groups with four hierarchical levels: alpha, beta, delta, and omega. Alpha wolves are the leaders of the pack and make decisions. Beta wolves act as advisors to the alpha wolves and help alpha wolves in making decisions. Delta wolves act as guardians, watchers, and hunters. Delta wolves also control omega wolves by obeying alpha and beta wolves. Omega wolves must obey every other wolf. In the gray wolf modeling, the hunting behavior of the pack of wolves while surrounding the prey has been formulated as follows [39].

$$\vec{X}(t+1) = \vec{X}_{\mathrm{p}}(t) + \vec{A} \cdot \vec{D} \tag{2}$$

$$\vec{D} = |\vec{C} \cdot \vec{X}_{\mathrm{p}}(t) - \overline{X}(t)| \tag{3}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{4}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{5}$$

$$a = 2 - 2 \times \left(\frac{Current\_iteration}{Max\_iteration} \right) \tag{6}$$

where A and C are the vectors of coefficients, X_{p} is the vector of hunting position, X is the position of wolves in dth-dimension, and d is the number of features. The variable t is the number of iterations, r_1 and r_2 are the vector of random numbers in range of [0,1] and vector a is a convergence factor that decreases linearly from 2 to 0. In this process, alpha wolves are shown as the best solution, and the positions of beta and delta wolves are considered additional knowledge. Therefore, as the three best solutions are obtained in one iteration, other wolves (e.g., Omega) have to change their positions in the decision space according to the best solutions. The above formulas for alpha, beta, and delta wolves are calculated separately with Eq. 7. Due to the continuous space of spatial variables, the above formulas are not suitable for feature selection strategy, and by changing them to a binary form, they can be used for feature selection, as introduced in [40]. Hybrid learning techniques

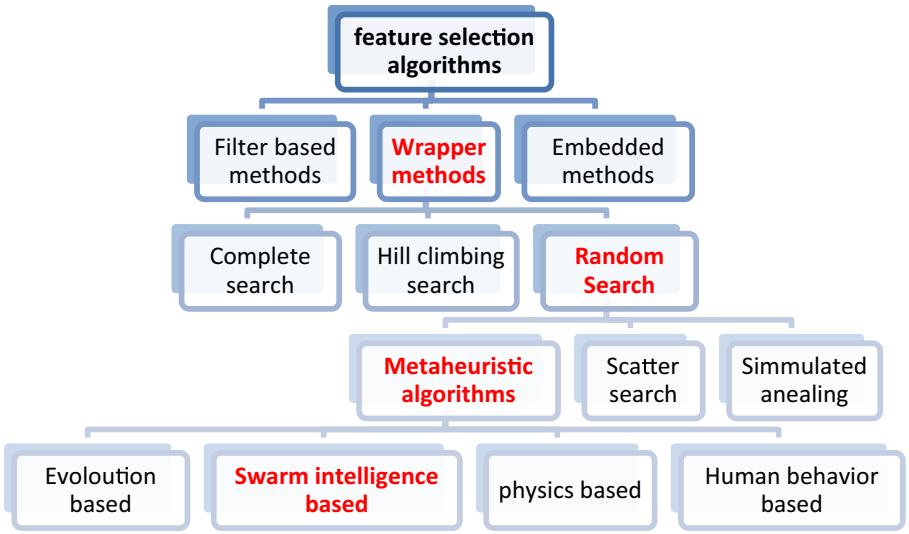

Fig. 3 Feature selection methods taxonomy

were inspired in this research, and genetic algorithm operators were embedded into this algorithm to enhance binary gray wolf algorithm. One of the main problems of the binary gray wolf algorithm is using a parameter to create a balance between exploration and extraction. In this research, two methods were used to improve this algorithm. In the classical GWO method, the parameter a has a linear behavior, so it tends to be explored at the beginning of the algorithm execution, and, during the iterations, it tends to more exploit than explore. By converting the linear behavior of this variable from linear to nonlinear, an acceptable balance between exploration and extraction can be created [41]. The following formula is used to update this variable in each iteration.

$$a = 2 - 2\left(\frac{Current\_iteration}{Max\_iteration}\right)^2 \tag{7}$$

In the next step, beta wolves participate more in coordinating pals toward the goal. In this way, the genetic algorithm operators are inferred in selecting parents, and two new children are produced by alpha and beta using the crossover operator technique [42]. According to the merit of children that is calculated using the cost function, they are either added to the population instead of the worst delta wolves or do not affect decision making. Indeed, the crossover operator has donated a second chance to a better position in the pack. This method accelerates convergence speed and performs a better search strategy. In this article, due to the large number of features, a uniform crossover was used to produce children, being empirically more effective than the single and multi-point crossover. In uniform crossover, each feature's bit was selected with an equal probability between parents (delta and beta wolf). The proposed GWOGA pseudo-code is described below.

Algorithm 2. Pseudo code for proposed GWOGA

Input: Population Size, Problem Size
Output: Best Particle (Alpha)

1. ***Initialization***
   Initialize A, a and C
   Randomly Initialize wolves' population
2. ***Calculate fitness value for population***
3. ***Attain Alpha, Beta and Gamma in population***
4. ***while (iteration < MaxIteration)***
   Update the position using Eq. 2
   Select Alpha and Beta and change individuals using uniform crossover operation
   Evaluate Alpha and Beta
   Update A, a and C
   Evaluate All individuals
   Update the position of three best Agents (Alpha, Beta, Delta)
   Iteration = iteration +1
5. ***end while***
6. ***return Alpha***

Random forest (RF) algorithm was used to evaluate the feature subset. RF is a user-friendly machine learning algorithm that often obtains very good results without tuning meta-parameters. RF has been extremely successful as a general-purpose classification and regression method. This method includes a group of decision trees that has tuned the parameters and construct trees using the bagging technique. The optimization results of this technique directly depend on the correlation of decision trees, with the error decreasing with an increase in the correlation of the trees. This technique has utilized major voting technique for the final classification result [5]. RF algorithm is highly prone to overfitting that makes the model incapable of classifying new or unseen data as well as the training data. In order to overcome the overfitting problem, 10-fold Cross-validation is utilized in the algorithm that divides the data into ten categories, nine parts are considered for training and one part for testing. This tactic is repeated ten times so that each category is considered test data once [43].

3.4 Proposed quadratic nonlinear regression model

In this section, the proposed model for intrusion detection systems is presented. After cleaning the data and selecting the most important features, quadratic nonlinear regression modeling was used to formulate the network behavior and detect legitimate and attack traffics. Regression methods are widely used in machine learning processing, but some simple versions of regression like linear regression cannot be used in sophisticated problems like network behavior modeling. Regression is a technique for investigating and discovering the relationship between attributes (x) and a dependent variable or class (y). The regression model presented in this research is shown in Eq. 8. The coefficients of the proposed model were optimized using a new and prominent meta-heuristic algorithm.

$$f(x) = \sum_{i=1}^{N} \alpha_i x_i^2 + \sum_{j=1}^{N} \sum_{k=j+1}^{N} \beta_{ij} x_i x_j + \varepsilon \tag{8}$$

$$G(x) = \frac{1}{1 + e^{-f(x)}} \tag{9}$$

$$\text{Rule}: \begin{cases} 1 \text{ if } G(x) >= 0.5 \\ 0 \text{ if } G(x) < 0.5 \end{cases}$$

where α and β are the coefficients of the model optimized by the HGS and ε is the momentum term, x is the values of the selected features, and N is the number of features. After simplifying the above equation, the total number of generated coefficients is equal to $N + \binom{N}{2} + 1$ which is optimized by a meta-heuristic algorithm. In Eq. 9, the sigmoid function is used to control the output range of the proposed model. If the value of this function is greater than 0.5, the output is 1, and if it is less than 0.5, the output value is considered equal to zero. In the proposed model, hunger games search (HGS) meta-heuristic algorithm was utilized for solving the mentioned optimization problem. HGS algorithm was introduced in 2021 as a swarm intelligence algorithm inspired by the behavior of animals during starvation [44]. Different researches have proven that hunger is a reliable persuasive force for movement, learning, and food search, and stimulates animals to change their existing conditions to a more stable state. The behavior of organisms according to the concept of hunger is used to create an adaptive weight, also using the consequence of hunger in each step of the search process to reach the food source. In this way, a higher relative weight of hunger brings higher survival and food acquisition probability. Due to the limited food source, food acquisition among hunger animals is converted to a critical game. Animals are divided into two categories: cooperative and non-cooperative, based on their power and starvation level. According to Eq. 10, when the animals behave cooperatively, they follow GAME1, and when they are non-cooperative, they behave according to the other two games. The exploration and extraction process of this algorithm is modeled using the following equations.

$$X_{(t+1)} = \begin{cases} Game_1 = X_t \cdot (1 + rand(1)) & r1 < l \\ Game_2 = W_1 \cdot X_{\text{best}} + R \cdot W_2 \cdot |X_{\text{best}} - X_t| & r1 > l, r2 > E \\ Game_3 = W_1 \cdot X_{\text{best}} - R \cdot W_2 \cdot |X_{\text{best}} - X_t| & r1 > l, r2 < E \end{cases} \tag{10}$$

$$R = 2 \times Shrink \times rand - Shrink \tag{11}$$

$$E = \text{sech}(|f_{(i)} - f_{\text{best}}|) \tag{12}$$

$$Shrink = 2 \times \left(1 - \frac{CurrentIteration}{MaxIteration}\right) \tag{13}$$

$$\text{Sech}(x) = \frac{2}{e^x + e^{-x}} \tag{14}$$

in which X_t and X_{t+1} are the updated position and the current position of the animals. $rand(1)$, $r1$, and $r2$ are the random function with normal distribution and random values in the interval [0,1], respectively. The position X_{best} is the best particle obtained so far, and the variables R and E are responsible for controlling the range and variations, which are calculated using formulas 11 and 12. The variable $Shrink$ decreases gradually with each iteration and is calculated by Eq. 13, and the Sech, $f(i)$ and f_{best} demonstrate hyperbolic function, the value of the objective function of each particle, and the best value of the objective function so far, respectively. The adaptive weight values for the hunger power in particles were

modeled according to Eqs. 15 and 16. The variable N represents the size of the population of particles, *SumHungry* is the sum of the hunger feeling of the particles, *UB* and *LB* are the upper and lower bounds of the particles, f_{worst} is the worst

Hybrid meta-heuristic algorithms combine the best operators from different meta-heuristic algorithms and create a new advanced algorithm. The crossover and mutation operators in the genetic algorithm can do this well [46]. The pseudo-code of the proposed algorithm is shown below.

Algorithm 3. Pseudo code for proposed HGSGA

Input: Population Size, Problem Size
Output: Best Particle (X(1))
1. ***Initialization the l, SumHungry, N and MaxIteration parameters***
2. ***Initialize $X_{(i)}$ randomly in range of [-1,1]***
3. ***Evaluate population***
4. ***While (iteration < MaxIteration)***
   Update the f_{best}, f_{worst}, X_{best}
   Calculate hungry_level using Eq. 17
   Calculate W_1 and W_2 using Eq 15 and 16
   Foreach individuals:
   Calculate R and E using Eq. 11 and 12
   Update the position using Eq. 10
   End For
   Select Parents using Roulette Wheel and change individuals using uniform crossover operation
   Evaluate Childs and population using Eq. 8 and 9
   Select N individual for next iteration
   Iteration = Iteration + 1
5. ***End while***
6. ***Return X(1)***

solution of the objective function so far, and l is a constant value. According to Eq. 17, when the particle fitness function is equal to the best fitness function, it is no longer hungry and takes zero value. Otherwise, a new hunger value (H) is added to the current hunger level. Therefore, the hunger level of each particle is different.

$$w_1(i) = \begin{cases} hungry\_level_{(i)} \cdot \frac{N}{SumHumgry} \times r4 & r3 < l \\ 1 & r3 > l \end{cases} \tag{15}$$

$$w_2(i) = (1 - \exp(-\left|hungry\_level_{(i)} - SumHungry\right|)) \times r5 \times 2 \tag{16}$$

$$hungry\_level_{(i)} = \begin{cases} 0 & f_{(i)} == f_{\text{best}} \\ hungry\_level_{(i)} + H & f_{(i)} != f_{\text{best}} \end{cases} \tag{17}$$

$$H = \begin{cases} LH \times (1+r) & TH < LH \\ TH & TH >= LH \end{cases} \tag{18}$$

$$TH = \frac{f_{(i)} - f_{\text{best}}}{f_{\text{worst}} - f_{\text{best}}} \times r6 \times 2 \times (ub - lb) \tag{19}$$

However, HGS tends to get stuck in local optima in complex optimization problems as a consequence of premature convergence. In this research, this algorithm was combined with the genetic algorithm operators (HGSGA) to improve the performance of HGS algorithm in the exploration and exploitation phases, converge the optimal solution rapidly, and maintain diversity in the population [45].

Generally, the proposed framework focused on hybrid meta-heuristic algorithms as a solution to intrusion detection strategy by transforming network behavior into an optimization problem. First, the CICIDS 2017 dataset was cleaned, balanced and normalized in the preprocessing step, and then, the standard dataset was embedded into wrapper algorithm. GWOGA algorithm identified the best feature subset with an ensemble evaluation method called random forest. In the last step, nonlinear quadratic regression was used to model the behavior of the network, and the model coefficients were optimized using one of the newest and most powerful meta-heuristic algorithms, HGS. To improve the efficiency and maintain diversity in population, the main GA operators were embedded into HGS. The presented framework for intrusion detection exerted efficient methods and strategies on the dataset and utilized the latest and most powerful algorithms. In order to evaluate the performance of the proposed framework, the simulation results are explained in the next section.

4 Simulation results

In this section, the performance of the proposed framework is investigated based on simulation results. In the simulation, Python 3.9 and MATLAB R2020b programming language frameworks were utilized in this research. All of the simulations were run on a 10th-generation i5 CPU and 16 GB of RAM.

4.1 Proposed model results

The results of the feature selection and detection method are shown separately for each type of attack. In this research, the wrapper method based on the hybrid GWOGA search method and the RF ensemble classifier was utilized for subset evaluation. The main aim of feature selection is to create a subset with minimal features and maximum efficiency. The selected feature subsets were also examined by the decision tree for a more accurate evaluation of the subset. The proposed model was trained by the selected features and evaluated to detect different attacks in the next step. The evaluation metrics in this research were precision, accuracy, recall, and F-measure. Accuracy measured the model's capability to recognize the attack and legal classes correctly, and precision measured the model's ability not to detect the legal class as an attack. Recall indicated all positive values, i.e., how many were predicted positive, and the F-measure showed the weighted average of accuracy and recall.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{20}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{21}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{22}$$

$$F\text{-Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{23}$$

The parameters used in Eqs. 20–23 are as follows.

- TP: the number of Attack samples classified as Attack.
- FP: the number of Normal samples classified as Attack.
- TN: the number of Normal samples classified as Normal.
- FN: the number of Attack samples classified as Normal.

4.1.1 Port scan attack

Port scan attack is one of the most prevalent network attacks in the world. In this attack, open ports are compromised by intruders with the aim of arbitrary data transferring or information sniffing. This attack is considered a backdoor for more damage to the intruders. Table 4 shows the results of Port scan attack feature selection. As shown, only 11 features were selected and evaluated.

After evaluating the subset of features and ensuring efficiency, the selected features were embedded into the nonlinear regression model and then the training and testing results were obtained. Figure 4 shows the training trend of the proposed method that converged to the optimal solution. In different attacks, right line charts depicted the best answer, average answer, and worst answer for each iteration. The chart on the left shows the best solution trend obtained in training process, which confirming the stability of the proposed algorithm. The mean square error was used as the objective function to evaluate the algorithm's performance (Eq. 24), where n, Y and $\hat{Y}$ are the number of training data samples, actual value of the class and $\hat{Y}$ predicted value, respectively.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{24}$$

4.1.2 Brute force attack

Another attack in the CICIDS dataset is Brute force attack. This attack tries to access the password, login credentials, and encryption keys using trial and error to penetrate into the network as a legitimate user. Brute force attacks on SSH and FTP protocols were collected in this dataset. The results obtained for detection of brute force attack are shown in Table 5 and Fig. 5.

4.1.3 DDoS attack

This attack has been one of the most common attacks in recent years, which disrupts network accessibility to disable services. In this attack, the intruder usually uses trusted third-party servers for the attack so that the intruder's information remains untraceable [47]. The results obtained for the feature subset and detection of this attack are shown in Table 6 and Fig. 6.

4.1.4 Botnet attack

Botnet attacks aim to infect computer systems in the network and turn them into controllable bots under the intruder's control. In recent years, attacks such as Mirai and Gafgyt, which are introduced as botnet attacks, have grown rapidly and caused irreparable damage to various networks such as IoT. This attack can be a prerequisite for many other attacks and intrusions, such as DDoS [48]. Table 7 shows the results of the selected features for Botnet detection, and Fig. 7 shows the training of the proposed model to detect this attack.

4.1.5 Infiltration

This attack is a malicious activity that penetrates and damages the network and manipulates the software in the network. One of the severely imbalanced data was dedicated to this attack. After using the Smote Tomek method and data preparation,

Table 4 Feature selection results for Port scan

Feature subset	Destination Port, Flow Duration, Fwd packet length min, Bwd Packet Length Min, Flow Packets/s, Flow IAT Max, Min Packet Length, Packet Length Std, ACK Flag Count, Fwd Header Length.1, Subflow Fwd Packets (11 features among 79)					
	TP rate	FP rate	Precision	Recall	F-measure	Class
	1.00	0.00	1.00	1.00	1.00	0
	1.00	0.00	1.00	1.00	1.00	1
Weighted Avg	1.00	0.00	1.00	1.00	1.00	

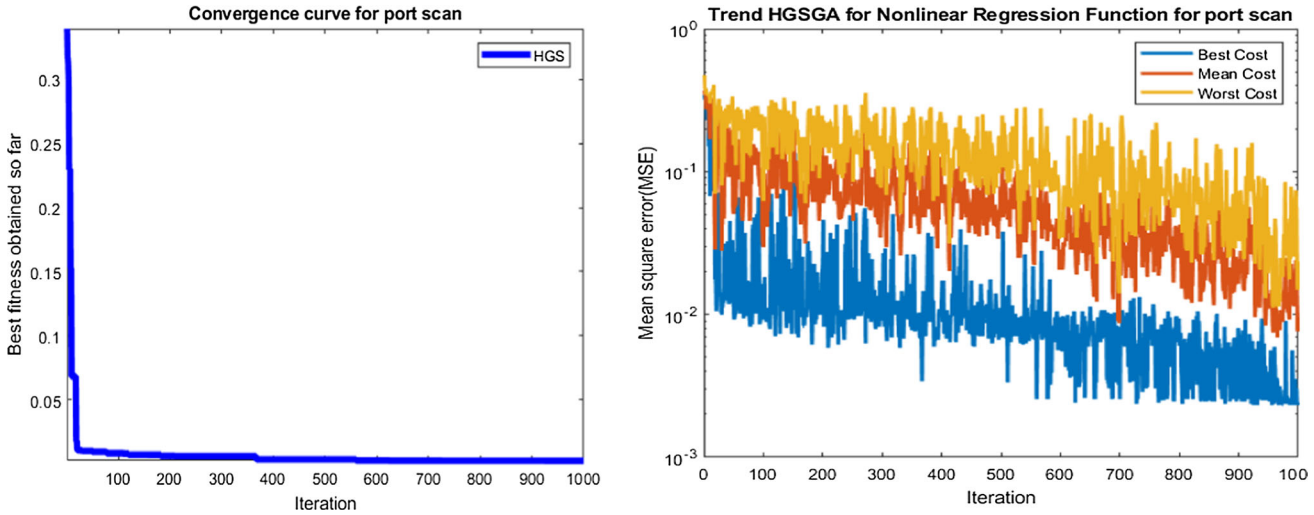

Fig. 4 HGSGA training for port scan attack

Table 5 Feature selection results for Brute Force

Feature subset	Destination Port, Bwd Packet Length Min, Fwd IAT Max, Bwd IAT Mean, Bwd IAT Max, Fwd Packets/s, Bwd Packets/s, SYN Flag Count, Down/Up Ratio, Init_Win_bytes_backward (10 features among 79)					
	TP rate	FP rate	Precision	Recall	F-measure	Class
	1.00	0.003	1.00	1.00	1.00	0
	0.997	0.000	1.00	0.997	0.999	1
Weighted Avg	1.00	0.003	1.00	1.00	1.00	

it entered the feature selection and intrusion detection phase. The results are shown in Table 8 and Fig. 8.

As mentioned in Sect. 3, after training the model, test data were prepared for performance evaluation, and then, the corresponding class of each feature was predicted. To ensure the stability of the obtained result and random nature of meta-heuristic algorithms, each part of the algorithm was executed 5 times, and the results were averaged. One of the vital criteria in performance evaluation is test accuracy, which is separately shown for all attacks in Table 9.

As shown in Table 8, the proposed method for different types of attacks in the CICIDS dataset provided an efficient performance, and the proposed feature selection method, selecting the minimum feature, was also able to reach high accuracy. The high-quality feature subset consisting of minimum features could considerably improve the complexity and time-consuming training process. According to Figs. 4, 5, 6 and 7, the line chart of the training process shows that the algorithm represented both small movements and large jumps, confirming suitable exploration and exploitation capabilities of the proposed algorithm. In much recent research, the class imbalance problem of the dataset has not been addressed in the preprocessing step, which directly affects the performance and as a result the obtained results are not definitively reliable. Here, before the feature selection phase, Smote Tomek was utilized to make a balance in classes by combining oversampling and under-sampling techniques. In order to prove the superiority of the proposed method in term of efficiency and detection accuracy,

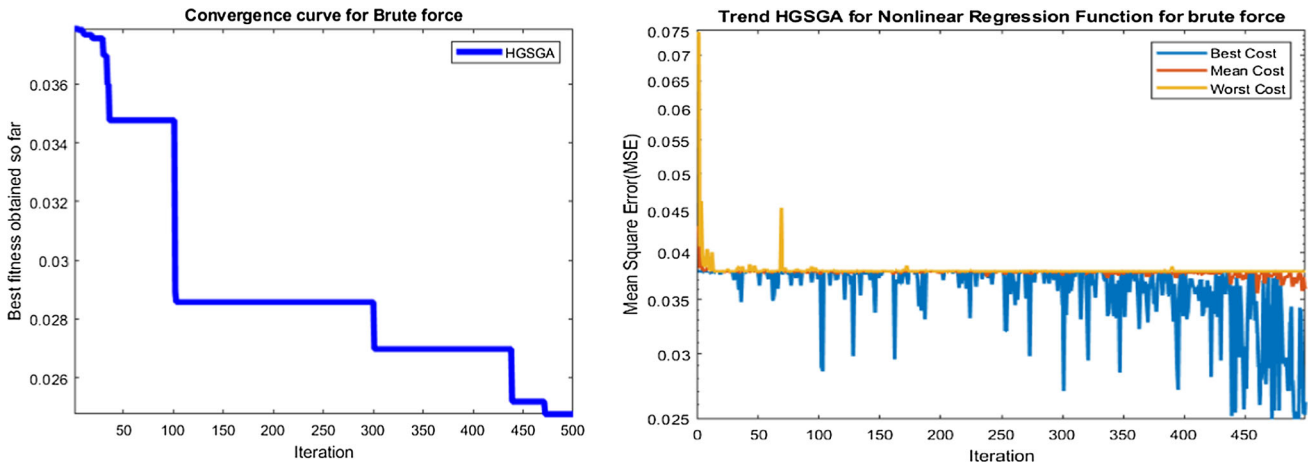

Fig. 5 HGSGA training for Brute force attack

Table 6 Feature selection results for DDoS/Dos

Feature subset	Destination port, Bwd Packet length mean, Fwd IAT std, PSH flag count, Average Packet Size (5 features among 79)					
	TP rate	FP rate	Precision	Recall	F-measure	Class
	0.999	0.000	1.00	0.999	0.999	0
	1.00	0.001	0.999	1.00	1.00	1
Weighted Avg	1.00	0.000	1.00	1.00	1.00	

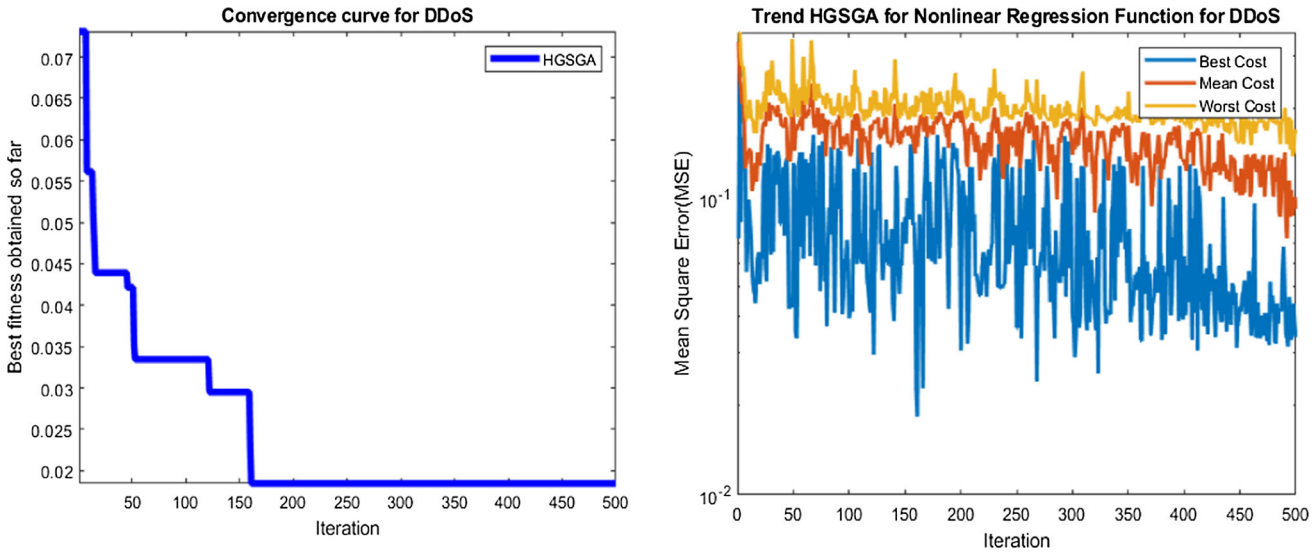

Fig. 6 HGSGA training for DDoS attack

Table 7 Feature selection results for Botnet

Feature subset	Destination Port, Flow Duration, Fwd Packet Length std, Flow Packets/s, Flow IAT std, Fwd IAT Mean, Bwd IAT Std, Fwd Header Length, Bwd Header Length, Init_win_bytes_forward, Init_win_bytes_backward (11 features among 79)					
	TP Rate	FP Rate	Precision	Recall	F-measure	Class
	1.00	0.015	1.00	1.00	1.00	0
	0.985	0.000	0.987	0.999	0.986	1
Weighted Avg	1.00	0.015	1.00	1.00	1.00	

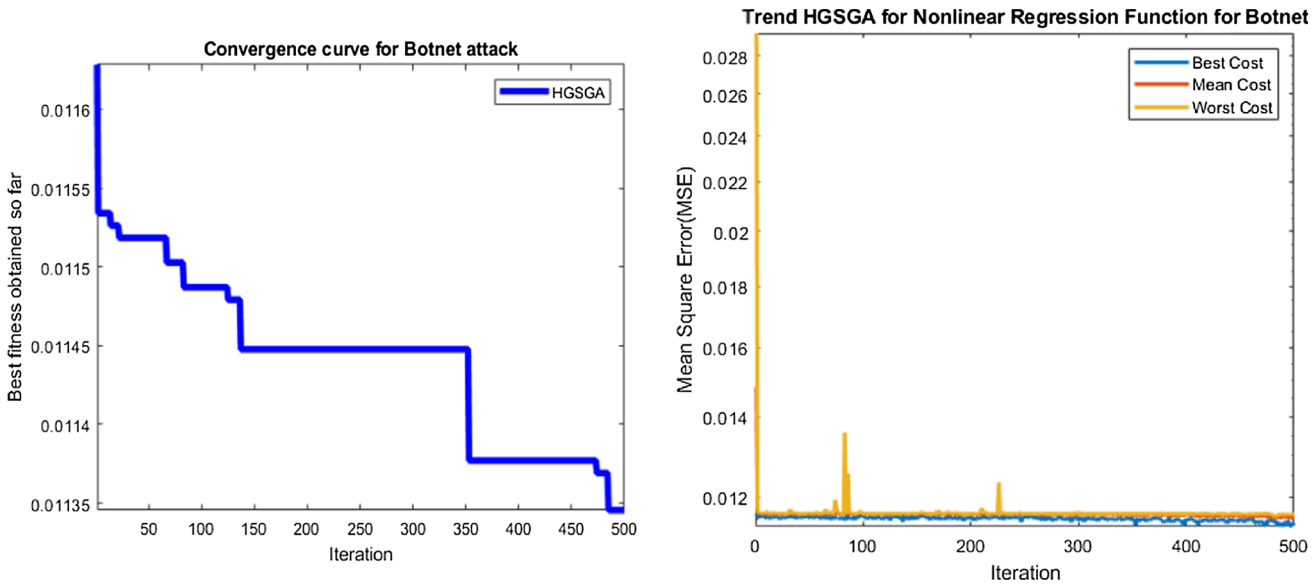

Fig. 7 HGSGA training for Botnet attack

Table 8 Feature selection for Infiltration

Feature subset	Destination port, Flow Duration, Total length backward packets, Backward packet length max, Flow Bytes, Flow Packets, Flow IAT mean, Flow IAT std, Flow IAT max, Flow IAT min, Forward packet/s, Backward packet/s, Packet length min, RST flag count, Init_forward_win_bytes, Forward segment size min, idle min (17 features among 79)					
	TP rate	FP rate	Precision	Recall	F-measure	Class
	0.999	0.000	1.00	0.999	1.0.999	0
	1.00	0.001	0.999	1.00	1.00	1
Weighted Avg	1.00	0.000	1.00	1.00	1.00	

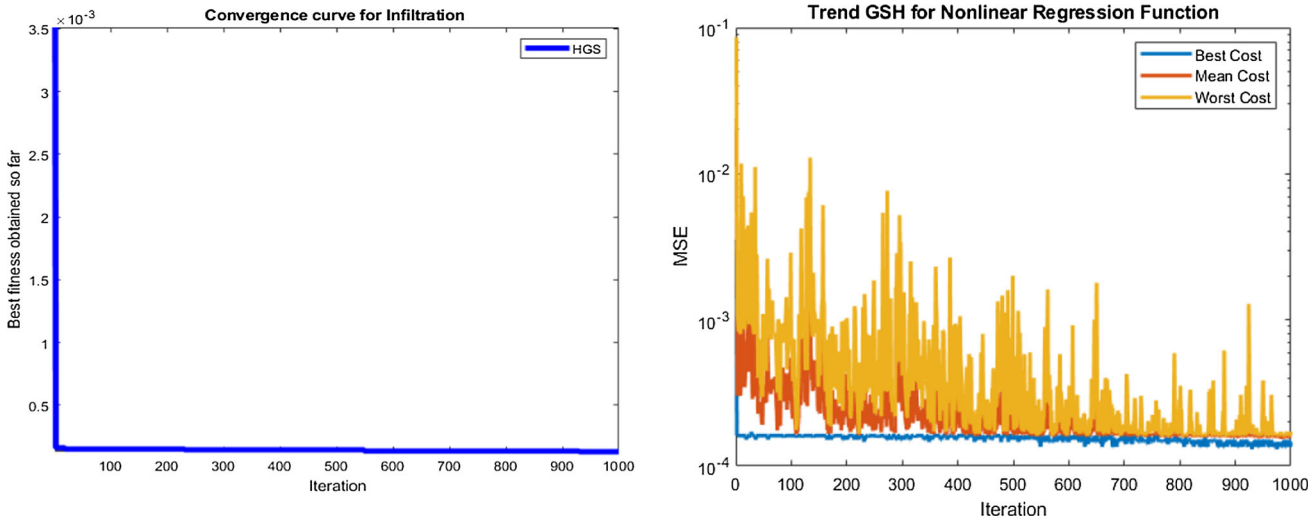

Fig. 8 HGSGA training for Infiltration attack

Table 9 Test and train result of proposed method for different attacks

	Hybrid HGSGA		Classical HGS	
	Train accuracy	Test accuracy	Train accuracy	Test accuracy
DDoS	99.4766	99.039	98.1532	96.8648
Botnet	99.252	98.8654	99.122	98.8482
Brute force	98.5237	98.2679	98.2322	96.5375
Port scan	99.7677	99.7358	98.3956	98.144
Infiltration	99.987	99.9823	92.6772	82.7028

Table 10 The comparative analysis of the proposed method and related works

	Dataset	Attack type	Preprocessing	Feature selection strategy	Number of features	Detection strategy	Results (%)
[25]	VirusTotal	Destructive data	Compress data Data transformation	–	–	SVM approach of modified support vector	0.97
[20]	KDD-cup99	All attack	Data transformation	–	–	PSO-RF	96.5
	UNSW-NB15						96.3
[48]	N-BaIoT	Botnet	–	–	–	Local–Global best Bat Algorithm for Neural Network	90
[19]	CICIDS-2017	1. Multiclass	Random sampling (10%)	Double PSO	23	Deep belief network	1. Avg: 95.81
		2. Binary	Normalization				2. 99.91
[21]	1. NSL-KDD	All attack	Mapping	Grey wolf + PSO	Avg: 10.2	Decision Tree	1 .99.058
	2. UNSW-NB15		Transformation				2. 94.482
			Normalization				
[16]	NSL-KDD	DDoS	Nominal to numeric	Lion optimization algorithm	20	CNN	96
[27]	NSL-KDD	All attack	Encoding	GWO	12	SVM	96
			Normalization				
[49]	CICIDS-2017	Brute force	Numeralization	–	–	KNN	99.49
		XSS	Normalization				
		SQL injection					
Proposed method	CICIDS-2017	1. DDoS		GWOGA + RF	1. 5	HGSGA + nonlinear regression	1. 99.039
		2. Infiltration	Data cleaning		2. 17		2. 99.9823
		3. Brute Force	Smote Tomek		3. 10		3. 98.2679
		4.Port Scan	Normalization		4. 11		4. 99.7358
		5. Botnet			5. 11		5. 98.8654

the comparative analysis of the proposed model with recent related works was conducted, shown in Table 10. The utilization of antiquated datasets, low accuracy, unreliable results, and high-dimensional feature subset could be considered the main weaknesses of the related research.

5 Conclusion

Nowadays, the expansion of computer networks and their applications have made Internet attacks more sophisticated. In addition, common preventive strategies such as authentica-

tion, firewall, and antivirus are not sufficient enough against complex cyber-attacks. In this respect, artificial intelligence (AI)-based intrusion detection systems have been recently the subject of intense research. However, research has shown the proposed intrusion detection systems suffer from low accuracy and high false alarm rates. Antiquated datasets have been widely used to evaluate the proposed methods without considering the class imbalance. Considering the existing challenges and the complexity of network behavior, predicting and learning network behavior has turned into an NP-Hard problem. The present study offered a robust and efficient system for intrusion detection using meta-heuristic optimization techniques and quadratic nonlinear regression modeling. The proposed model was evaluated using the CICIDS dataset, which was preprocessed before use. In the preprocessing step, the dataset was cleaned, and the problem of unbalanced classes was solved using the new SMOTE Tomek link method. In the next step, redundant features were removed using the hybrid GWOGA-RF algorithm, and a subset of features suitable for each attack was selected. The selected subsets included the smallest number of features and the highest efficiency. In the last step, the nonlinear regression model was trained using a subset of features and optimized using a new meta-heuristic algorithm called HGS. The intersection operator in the genetic algorithm (GA) was combined with this algorithm to enhance the performance of HGS algorithm and reach a faster convergence. The results showed that the proposed system was able to detect different attacks with high accuracy and outperform other models presented in other studies. The results of this research can be summarized as follows.

- Smote Tomek link method was applied to solve the imbalance problem of the CICIDS dataset.
- The wrapper method was applied, and a feature subset with a small number of features and high accuracy was developed. This method benefited from Grey Wolf Optimization (GWO) algorithm to search for the subset and RF algorithm to evaluate the subsets.
- The improved GWO was combined with GA to enhance the performance of the search method in Wrapper algorithm. The results showed a very good performance with an average of 11 function features.
- Turning the problem into an NP-Hard optimization problem, a quadratic nonlinear regression model was developed to model the behavior of the network.
- A new hybrid meta-heuristic algorithm was used for optimization. This algorithm could detect common mentioned attacks with an average test accuracy of 99.17%.

Author contribution SM supervised the research. MB designed and implemented the framework. All authors discussed the results and contributed to the final manuscript.

Funding The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

Availability of data and materials The data that support the findings of this study are available on request from the corresponding author.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval We would confirm that this paper has not been published nor submitted for publication elsewhere. We confirm that we have read, understand and agreed to the submission guidelines, policies and submission declaration of the journal.

References

1. Jose, N., Govindarajan, J.: DOMAIN-based intelligent network intrusion detection system. In: Smys, S., Balas, V.E., Kamel, K.A., Lafata, P. (eds.) Inventive Computation and Information Technologies, pp. 449–462. Springer, Berlin (2022)
2. Khan, M.A., Kim, J.: Toward developing efficient Conv-AE-based intrusion detection system using heterogeneous dataset. Electronics **9**(11), 1771 (2020)
3. Khraisat, A., et al.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity **2**(1), 1–22 (2019)
4. Liu, H., Lang, B.: Machine learning and deep learning methods for intrusion detection systems: a survey. Appl. Sci. **9**(20), 4396 (2019)
5. Thakkar, A., Lohiya, R.: A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges. Arch. Comput. Methods Eng. **28**(4), 3211–3243 (2021)
6. Janiesch, C., Zschech, P., Heinrich, K.: Machine learning and deep learning. Electron. Mark. **31**(3), 685–695 (2021)
7. Zainab Ali, A., Ismael, K., Karan, A.: Challenges and future directions for intrusion detection systems based on AutoML. Mesop. J. CyberSecur. **2021**, 16–21 (2021)
8. Hussain, K., et al.: On the exploration and exploitation in popular swarm-based metaheuristic algorithms. Neural Comput. Appl. **31**(11), 7665–7683 (2019)
9. Kumar, G.: An improved ensemble approach for effective intrusion detection. J. Supercomput. **76**(1), 275–291 (2020)
10. Dhiman, G.: ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. Eng. Comput. **37**(1), 323–353 (2021)
11. Singh, P., Kottath, R.: An ensemble approach to meta-heuristic algorithms: comparative analysis and its applications. Comput. Ind. Eng. **162**, 107739 (2021)
12. Ajdani, M., Noori, A., Ghaffary, H.: Providing a consistent method to model the behavior and modelling intrusion detection using a hybrid particle swarm optimization-logistic regression algorithm. Secur. Commun. Netw. **2022** Article ID 5933086, 1–7 (2022)
13. Ahmad, Z., et al.: Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans. Emerg. Telecommun. Technol. **32**(1), e4150 (2021)

14. Nazir, A., Khan, R.A.: Network intrusion detection: taxonomy and machine learning applications. In: Maleh, Y., Shojafar, M., Alazab, M., Baddi, Y. (eds.) Machine Intelligence and Big Data Analytics for Cybersecurity Applications, pp. 3–28. Springer, Berlin (2021)
15. Gulghane, S., et al.: A survey on intrusion detection system using machine learning algorithms. In: *International Conference on Innovative Data Communication Technologies and Application*. Springer (2019)
16. Arivudainambi, D., Varun Kumar, K.A., Sibi Chakkaravarthy, S.: LION IDS: A meta-heuristics approach to detect DDoS attacks against Software-Defined Networks. Neural Comput. Appl. **31**(5), 1491–1501 (2019)
17. Ajdani, M., Ghaffary, H.: Improving network intrusion detection by identifying effective features based on probabilistic dependency trees and evolutionary algorithm. Clust. Comput. **25**(5), 3299–3311 (2022)
18. Pandey, J.K., et al.: A Metaheuristic autoencoder deep learning model for intrusion detector system. Math. Probl. Eng. **2022**, 1–11 (2022)
19. Elmasry, W., Akbulut, A., Zaim, A.H.: Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Comput. Netw. **168**, 107042 (2020)
20. Ajdani, M., Ghaffary, H.: Introduced a new method for enhancement of intrusion detection with random forest and PSO algorithm. Secur. Privacy **4**(2), e147 (2021)
21. Alzubi, Q.M., et al.: Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization. Expert Syst. Appl. **204**, 117597 (2022)
22. Zhou, Y., et al.: Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput. Netw. **174**, 107247 (2020)
23. Singh, S., Singh, A.K.: Detection of spam using particle swarm optimisation in feature selection. Pertan. J. Sci. Technol. **26**(3), 1355–1372 (2018)
24. Quincozes, S.E., et al.: An extended assessment of metaheuristics-based feature selection for intrusion detection in CPS perception layer. Ann. Telecommun. **77**(7-8), 457–471 (2022)
25. Ajdani, M., Ghaffary, H.: Design network intrusion detection system using support vector machine. Int. J. Commun. Syst. **34**(3), e4689 (2021)
26. Otair, M., et al.: An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks. Wirel. Netw. **28**(2), 721–744 (2022)
27. Safaldin, M., Otair, M., Abualigah, L.: Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. J. Ambient. Intell. Humaniz. Comput. **12**(2), 1559–1576 (2021)
28. Imran, M., et al.: Intrusion detection in networks using cuckoo search optimization. Soft Comput. **26**(20), 10651–10663 (2022)
29. Vamsi Krishna, K., et al.: A detailed analysis of the CIDDS-001 and CICIDS-2017 datasets. In: Ranganathan, G., Bestak, R., Palanisamy, R., Rocha, Á. (eds.) Pervasive Computing and Social Networking, pp. 619–638. Springer, Berlin (2022)
30. Liu, Y.-W., et al.: Optimal scheduling of combined cooling, heating, and power microgrid based on a hybrid gray wolf optimizer. J. Ind. Prod. Eng. **39**(4), 277–292 (2022)
31. Panigrahi, R., Borah, S.: A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. Int. J. Eng. Technol. **7**(3.24), 479–482 (2018)
32. Kaur, S., Singh, M.: Hybrid intrusion detection and signature generation using deep recurrent neural networks. Neural Comput. Appl. **32**(12), 7859–7877 (2020)
33. Tarawneh, A.S., et al.: Stop oversampling for class imbalance learning: a review. IEEE Access **10**, 47643–47660 (2022)
34. Elreedy, D., Atiya, A.F.: A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. Inf. Sci. **505**, 32–64 (2019)
35. Naeemullah, K., Ismael, K., Elika, D.: Improved feature selection method for features reduction in intrusion detection systems. Mesop. J. CyberSecur. **2021**, 9–15 (2021)
36. Agrawal, P., et al.: Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019). IEEE Access **9**, 26766–26791 (2021)
37. Moradi, P., Gholampour, M.: A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Appl. Soft Comput. **43**, 117–130 (2016)
38. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
39. Al-Tashi, Q., et al.: Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access **7**, 39496–39508 (2019)
40. Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary grey wolf optimization approaches for feature selection. Neurocomputing **172**, 371–381 (2016)
41. Ahmadi, R., Ekbatanifard, G., Bayat, P.: A modified grey wolf optimizer based data clustering algorithm. Appl. Artif. Intell. **35**(1), 63–79 (2021)
42. Singh, A.N., et al.: A comparative study of four genetic algorithm-based crossover operators for solving travelling salesman problem. In: Kumar, R., Singh, V.P., Mathur, A. (eds.) Intelligent Algorithms for Analysis and Control of Dynamical Systems, pp. 33–40. Springer, Berlin (2021)
43. Wong, T.-T., Yeh, P.-Y.: Reliable accuracy estimates from k-fold cross validation. IEEE Trans. Knowl. Data Eng. **32**(8), 1586–1594 (2019)
44. Yang, Y., et al.: Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. Expert Syst. Appl. **177**, 114864 (2021)
45. Li, S., et al.: A novel hybrid hunger games search algorithm with differential evolution for improving the behaviors of non-cooperative animals. IEEE Access **9**, 164188–164205 (2021)
46. Mahajan, S., Abualigah, L., Pandit, A.K.: Hybrid arithmetic optimization algorithm with hunger games search for global optimization. Multimed. Tools Appl. **81**, 28755–28778 (2022)
47. Kshirsagar, D., Kumar, S.: A feature reduction based reflected and exploited DDoS attacks detection system. J. Ambient. Intell. Humaniz. Comput. **13**(1), 393–405 (2022)
48. Alharbi, A., et al.: Botnet attack detection using local global best bat algorithm for industrial internet of things. Electronics **10**(11), 1341 (2021)
49. Maseer, Z.K., et al.: Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access **9**, 22351–22370 (2021)