Journal of Ambient Intelligence and Humanized Computing (2023) 14:7575–7592
https://doi.org/10.1007/s12652-023-04571-3

ORIGINAL RESEARCH

A novel network intrusion detection method based on metaheuristic optimisation algorithms

Reza Ghanbarzadeh[1] · **Ali Hosseinalipour**[2] · **Ali Ghaffari**[3]

Received: 7 February 2022 / Accepted: 15 February 2023 / Published online: 10 March 2023
© The Author(s) 2023

Abstract

The growing use of the Internet with its vulnerabilities has necessitated the adoption of Intrusion Detection Systems (IDS) to assure security. IDSs are protective systems that detect outsider infiltrations, unauthorised accesses and malfunctions occurring in computer networks. Intrusions can be detected and reported to the network administrator by IDSs using various pieces of information such as port scanning and irregular traffic detection. Intrusion detection is a classification problem, and identifying effective features is an essential aspect of classification methods. Standard methods used for classification are neural networks, fuzzy logic, data mining techniques and metaheuristics. One of the novel metaheuristic algorithms introduced to address optimisation problems is the Horse herd Optimisation Algorithm (HOA). This paper introduces a new approach on the basis of HOA for network intrusion detection. The new method uses horse behaviours in the herd to select effective features to detect intrusions and interactions between features. For the purpose of the new approach, HOA is first updated into a discrete algorithm using the floor function. The binarised algorithm is then converted into a quantum-inspired optimiser by integrating the concepts of quantum computing with HOA to improve the social behaviours of the horses in the herd. In quantum computing, Q-bit and Q-gate aid in striking a greater balance between the exploration and exploitation processes. The resulting algorithm is then converted into a multi-objective algorithm, where the objectives can be chosen from a set of optimal solutions. The new algorithm, MQBHOA, is then used for intrusion detection in computer networks, which is a multi-objective optimisation problem. For the classification, the K-Nearest Neighbour (KNN) classifier is applied. To evaluate the new algorithm's performance, two data sets, NSL-KDD (Network Security Laboratory—Knowledge Discovery and Data Mining) and CSE-CIC-IDS2018, are employed in which the network packets are classified into five categories: normal packets plus four intrusions packet types of Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L) and Probing (Prob). The new algorithm's performance was evaluated and compared with other well-known metaheuristic algorithms, and the influence of the parameters of the algorithm on the degree of intrusion was investigated. The results show a 6% more success rate in the average size of feature selection and the accuracy of classification in comparison with other evaluated algorithms. It also demonstrates an accuracy of 99.8% in detecting network intrusions compared to other methods.

Keywords Network intrusion detection · Metaheuristic · Horse herd optimisation algorithm · Feature selection · Classification

✉ Reza Ghanbarzadeh
Reza.Ghanbarzadeh@scu.edu.au

Ali Hosseinalipour
Ali.Hosseinalipour@yahoo.com

Ali Ghaffari
a.ghaffari@iaut.ac.ir

[1] Faculty of Science and Engineering, Southern Cross University, Gold Coast, Australia

[2] Department of Computer Engineering, Heris Branch, Islamic Azad University, Heris, Iran

[3] Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

1 Introduction

In recent years, network security has become more crucial to researchers than ever due to rapid computer network advancements and developments. Detecting intrusions is one of the primary methods for establishing security in any infrastructure (Dwivedi et al. 2021). Intrusion Detection Systems (IDSs), also called Network Intrusion Detection (NID) systems, are a type of application that analyse the traffic or requests in a network to identify intrusive activities such as unauthorised access, abuse and vulnerabilities created by

internal users and external attackers, and when it detects incoming network traffic is not permitted by network users or is the result of intruder activity, it will notify the network administrator immediately or display a customised response. A NID system consists of a number of tools and methods that monitor computer systems and network traffic while also analysing activities to look for potential system intrusions (Ozkan-Okay et al. 2021). Therefore, NID systems form a large part of the security infrastructure of organisations. An IDS could be used as hybrid, anomaly-based, or signature-based. In this respect, the application of intelligent machine learning approaches and expert systems to predict anomalies in computer networks is increasing. Figure 1 depicts a network structure with an IDS/NID system.

Numerous automatic techniques have been developed and introduced for NID in the literature; however, they do not provide 100% intrusion detection accuracy. Among all the proposed methods, machine learning algorithms have been the most successful (Tang et al. 2016). It is anticipated that employing an effective metaheuristic algorithm to select the most optimal input parameters can achieve a desirable outcome in terms of efficiency, detection time, and computing complexity of NID. The main two steps of metaheuristic techniques are feature selection and classification (Krishnaveni et al. 2021). In this study, the improved Horse herd Optimisation Algorithm (HOA) is used for feature selection, and K-Nearest Neighbour (KNN) is employed to carry out the classification phase in addressing the intrusion detection problem.

HOA is a recent nature-inspired metaheuristic introduced by MiarNaeimi et al. (2021). This algorithm demonstrates strong exploration and exploitation capabilities and can well achieve the optimal solution for large-dimensional problems. The behaviour of horses in their daily life is the foundation of HOA. Horse behaviours, including grazing, hierarchy, sociability, imitation, defence system, and roaming, at various ages, serve as the inspiration for the HOA method. Because of the numerous control factors based on the above behaviours, this algorithm performs exceptionally well in addressing complex problems with high dimensions.

This paper aims to develop a new NID method based on HOA. To achieve this goal, the continuous HOA is first converted into a discrete algorithm. Quantum computing is then integrated with the discrete HOA to improve horse movement and balance exploration and exploitation. The intrinsic qualities of quantum computing's Q-bit and Q-gate concepts aid in striking a greater balance between the search process' exploration and the exploitation properties (Srikanth et al. 2018). The resulting algorithm is then transformed into a multi-objective algorithm to tackle multi-objective problems. Finally, the Multi-objective Quantum-inspired Binary Horse herd Optimisation Algorithm (MQBHOA) is applied to select important features in the NID problem. The NSL-KDD and CSE-CIC-IDS2018 data sets are employed in implementing and evaluating the new method. KNN is used for the classification process in this study. KNN is a classifier based on supervised learning that employs proximity to producing classifications or predictions about the grouping of a single data point. Although it could be used for both classification or regression issues, it is commonly applied as a classification algorithm because it relies on the idea that similar points can be discovered close to one another. KNN is a non-parametric machine learning algorithm that does not require assumptions about the dataset. During the testing stage, the full training set is utilised. The decisions made by KNN are based on the entire training data. In comparison with the existing methods, the results indicate the introduced method's higher efficiency and accuracy.

To enhance the precision of detecting intrusion, studies in the literature have used a variety of methodologies, including data mining techniques, machine learning algorithms, and ensemble methods. Learning models are becoming more and more common for a number of applications, including machine vision, pattern recognition, and natural language processing, thanks to rapid improvements in computer processing units and the accessibility of public datasets. Based on the efficacy of the techniques, cyber security researchers are adopting this trend towards applying learning models for intrusion detection. The majority of the earlier machine learning-based techniques suffer from high attack detection rates and uncertainty. Various optimisation problems have been addressed by the development and introduction of numerous metaheuristic algorithms. Not all of those algorithms, though, are very effective. Many of them struggle with time complexity, others with significant computational complexity, some with the potential to become stuck in local optima, and so on. As various well-known test functions have benchmarked HOA at high dimensions, it is a unique, quick, robust, and reliable method that has none of the aforementioned problems and outperforms majority of the existing metaheuristic algorithms. In the analysis of eleven test functions, the algorithm surpasses current high-performance algorithms in high-dimensional spaces (Krishnaveni et al. 2021).

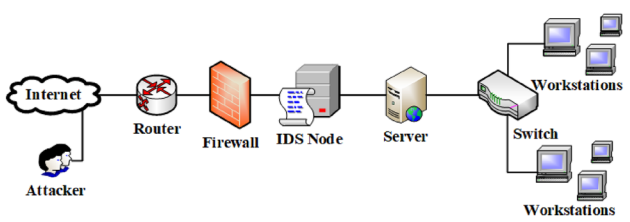

Fig. 1 Common structure of a network with IDS

This study contributes to the literature in the field in the following ways:

- HOA was originally a single objective algorithm; to address multi-objective optimisation problems, the algorithm was transformed into a multi-objective optimisation algorithm (Hosseinalipour and Ghanbarzadeh 2022a).
- To better balance exploration and exploitation processes, this study introduces a quantum-inspired version of the HOA algorithm.
- The improved HOA, MQBHOA, is used for feature selection along with the KNN classifier in detecting intrusions in computer networks and shows better accuracy, precision, sensitivity and F1-score compared to similar methods.
- The original HOA algorithm is a continuous optimisation algorithm; the current study also presents a discrete version of HOA to address discrete optimisation problems.

The remainder of this article is organised as follows. In Sect. 2, some of the previous solutions for intrusion detection will be discussed. Section 3 explains the background and preliminary where HOA will also be described. In Sect. 4, the new method will be presented, and Sect. 5 will present the evaluation results. Section 6 will discuss the limitation of the current study and future work, followed by the conclusion in Sect. 7.

2 Related works

Networking and the Internet have recently become an essential part of various applications. Today, security is one of the primary concerns of network designers (Abraham et al. 2021). Many existing network routing protocols use a single route to transmit data. The optimal route is selected from various criteria, such as information slope, the distance between the source and destination, and so on. Secure data transmission is one of the significant challenges in networking and the Internet, and it relies on selecting the optimal route. To deceive, attackers may send data in the wrong direction and damage the network in various ways; therefore, a procedure must be put in place to prevent such attacks. According to studies, the best way to ensure security is to consider it early in the design process (Karim et al. 2016). A proper security policy reduces the organisation's vulnerability. Intrusion detection is a problem of classification, and one of the main topics in classification is feature selection. The performance of a classifier does not depend linearly on the number of features; however, the functionality of the classifier will be compromised if the number of features is more than a specific amount. Feature selection for high-dimensional data not only reduces the cost and time of detection but also improves the classifier's performance (Chandrashekar and Sahin 2014).

Various intrusion detection solutions have been introduced in the literature (Bataghva 2017), and some of them are briefly discussed in this section. Due to their intelligent capabilities, machine learning-based NIDs have gained more and more attention in recent years. Artificial intelligence-based solutions can better identify variations of sophisticated network attacks than conventional signature-based approaches (Wu and Guo 2019). Traditional intrusion detection techniques, which prioritise rules, are insufficient to handle the growing complexity of network intrusion flows. As a result, several studies have used artificial intelligence techniques, such as deep learning and machine learning, in detecting intrusions in networks (Ahmad et al. 2021).

By integrating Convolutional Neural Networks (CNN) and gcForest, a multiple-layer representation learning model was introduced for NID by Zhang et al. (2019). Yu and Bian (2020) proposed a new deep learning method based on Few-Shot Learning (FSL) for NID. Six IDSs, based on machine learning techniques, were presented in the study by Karatas et al. (2020) using KNN, Random Forest (RF), Gradient Boosting, Adaboost, Decision Tree, and Linear Discriminant Analysis algorithms. Jiang et al. (2020b) proposed a NID algorithm that integrated hybrid sampling with a deep hierarchical network. A new Deep Neural Network (DNN) was presented by (Wu and Guo 2019) for NID that uses a hierarchical CNN and Recurrent Neural Network (RNN). Yang et al. (2020) applied a supervised adversarial variational autoencoder with regularisation and a DNN, in detecting known and unknown attacks in networks. A new deep learning method, called BAT, is introduced by Su et al. (2020) for NID, integrating bidirectional long short-term memory with the attention mechanism. Another NID method is proposed by Toldinas et al. (2021) that uses multistage deep learning image recognition. Wu et al. (2020) introduced a new NID technique on the basis of semantic re-encoding and deep learning. Binbusayyis and Vaiyapuri (2021) proposed an unsupervised deep learning single-stage method for NID that combines a one-dimensional convolutional autoencoder with a one-class Support Vector Machine (SVM) as a classifier.

The generalisation capability of NID systems based on classical machine learning methods is still insufficient and thus has a high false alarm rate. Therefore, there is a need for novel and robust methods to detect intrusion in networks with higher accuracy. Many studies have used heuristic and metaheuristic methods, as well as evolutionary and swarm intelligence in order to address the NID problem. For instance, based on the plants' biologically inspired response mechanism, Sharma et al. (2019) presented a model that can increase IDS performance in response to attacks. In the study by Hammad et al. (2020), NB, RF, J48

and ZeroR algorithms were used on UNSW-NB15 data set in order to detect intrusion. The results indicate that J48 and RF performed best in detecting intrusion, with 93.78% and 97.59%, respectively. Jiang et al. (2020a) introduced a new NID method on the basis of the PSO-Xgboost model. The authors first created a classification model based on Xgboost and then applied PSO to search for the optimal structure of Xgboost adaptively. A feature selection approach based on the wrapper was proposed in Nazir and Khan (2021) for NID, where Tabu is applied as a search method while RF is used as the learning algorithm. A novel method was introduced for NID, where an Artificial Neural Network (ANN) is employed as a learning technique. Additionally, the Grasshopper Optimization Algorithm (GOA) is applied for better and more accurate learning of ANNs to decrease the rate of errors in detecting intrusion (Moghanian et al. 2020). Deore and Bhosale (2022) utilised a deep long short-term memory based on chimp chicken swarm optimisation for the process of intrusion detection. A robust hybrid deep intrusion detection model based on whale optimiser using CNN features was introduced by Pingale and Sutar (2022). Another study proposed a hybrid NID model based on hybridisation bio-inspired metaheuristics such as Particle Swarm Optimisation (PSO), Multiverse Optimiser (MVO), Grey Wolf Optimiser (GWO), Moth-Flame Optimisation (MFO), Whale Optimisation Algorithm (WOA), FireFly Algorithm (FFA), and Bat Algorithm (BAT). To detect the generic attack, SVM, C4.5 (J48) decision tree, and RF classifiers were employed (Almomani 2021). Li et al. (2022) used an ANN to detect abnormal behaviour in a medical Internet of Things system, where the butterfly optimiser is used in selecting the optimal features for the process of learning in an ANN. With the integration of ensemble feature selection, GOA was applied by Dwivedi et al. (2021) for anomaly detection in an intrusion detection system.

As discussed earlier, there are a large number of classification and optimisation algorithms available; however, the reason for choosing HOA for this study is that this algorithm has the ability to address highly complex optimisation problems. Many other metaheuristics are only effective at handling non-complex situations and are unable to address complex optimisation problems. HOA performs exceptionally well in solving complicated high-dimensional problems because of the numerous control factors relevant to the behaviour of horses at various ages. Moreover, since harmonious relationships are established between the horses' movements, a higher exploration and exploitation level is achieved in this algorithm. The algorithm will be discussed in detail in the next section. HOA has been applied in addressing a large number of real-world and engineering applications, including security. For instance, Hosseinalipour and Ghanbarzadeh (2022a) employed an improved version of HOA in detecting email spam that shows substantially higher accuracy in the detection process. Evangeline and Rathika (2022) applied the horse herd optimisation algorithm for solving optimal power flow problems. In another study, HOA was used for solving fuel-constrained day-ahead scheduling of isolated nano-grid (NG) for five neighbouring homes (Basu and Basu 2021). Basu et al. (2022) applied HOA in solving convoluted economic dispatch problems. Mehrabi and Pashaei (2021) applied HOA to address discrete problems in biological data. HOA was used by Rajeshwari and Sughasiny (2022) to predict skin cancer's severity. Elmanakhly et al. (2022) used the discrete version of HOA in selecting the optimal subset of features for classification. HOA was used in another study for feature selection in text sentiment analysis (Hosseinalipour and Ghanbarzadeh 2022b).

The proposed approach is the combination of the improved version of HOA, MQBHOA, for feature selection and the KNN algorithm for classification, which can be used in detecting intrusions in computer networks. The main contribution is more on the feature selection using MQBHOA, which works well with KNN, as confirmed by the experiment results. The standard KNN classifier was used as the classification technique in this study, but many other classification algorithms can also come into the picture; however, KNN is one of the most popular ones and is used in many studies in the literature for classification purposes. KNN is a classifier based on supervised learning that uses proximity to produce classifications or predictions about grouping a single data point. Despite the fact that it could be used in classification or regression challenges, it is commonly applied as a classification algorithm because it relies on the idea that similar points can be discovered close to one another. KNN is a non-parametric machine learning algorithm that does not require any assumptions about the dataset to be used. During the testing stage, the full training set is utilised. The decisions made by KNN are based on the entire training data. Another supervised machine learning model, SVM, can be employed for classification and regression challenges. In SVM, each data item is represented as a point in n-dimensional space (where n is the number of available features), with each feature's value being the value of a certain coordinate. Next, the classification is performed by identifying the hyper-plane that effectively distinguishes the two classes. Support vectors are an individual observation's coordinates. A frontier that best separates the two classes (hyper-plane/line) is the SVM classifier. Another method is Naïve Bayes which is a classification method based on probability. In this classifier, it is assumed that, given the class label, attributes are conditionally independent of one another (Duda et al. 1995). Random Forest is another classification method that works based on a combination of tree predictors. This algorithm uses multiple binary decision trees that are built using a sample of data (Breiman 2001). In the current study, the classifier's performance is compared to various classifiers such as Naïve

Bayes, Random Forest, SVM and so on. It is believed that the MQBHOA-based feature selection works with classification techniques other than KNN as well. However, due to the high accuracy and performance of KNN in classification, its combination with MQBHOA works well in intrusion detection, as confirmed by the experiment results.

3 Background and preliminary

3.1 Horse herd optimisation algorithm

HOA, developed by MiarNaeimi et al. (2021), is a novel nature-inspired metaheuristic technique introduced to solve continuous optimisation problems. This technique simulates the day-to-day horse behaviours in the herd. A horse can live for up to 25–30 years (Krueger and Heinze 2008), and it shows various behaviours at different stages of its life. Based on their age, horses are classified into four groups in HOA: 0–5, 5–10, 10–15, and older than 15, which are denoted by δ, γ, β, and α, respectively. To simulate their social lives, this algorithm uses six general behaviours of horses of varying ages: grazing, hierarchy, sociability, imitation, defence mechanism and roam.

During each iteration of HOA, the movement of the horses is given by Eq. (1):

$$X_m^{Iter,AGE} = \overrightarrow{V}_m^{Iter,AGE} + X_m^{(Iter-1),AGE}, AGE = \alpha, \beta, \gamma, \delta \quad (1)$$

where $X_m^{Iter,AGE}$ represents the *mth* horse's position, $\overrightarrow{V}_m^{Iter,AGE}$ represents the *mth* horse's velocity vector, *AGE* represents the horse's age group, and *Iter* represents the current iteration. Every iteration must have a comprehensive response matrix to establish the horses' age. The matrixes are sorted on the basis of the best responses, with the top 10% of the horses chosen as α. The next 20%, 30% and 40% of the remainder of the horses in the matrix contain β, δ, and γ horses, respectively. The procedures of simulating the six above-mentioned behaviours are accomplished mathematically for detecting the velocity vector. Equation (2) is utilised to describe the motion vector of horses of various ages in each algorithm cycle (MiarNaeimi et al. 2021):

$$\begin{aligned}
\overrightarrow{V}_m^{Iter,\alpha} &= \overrightarrow{G}_m^{Iter,\alpha} + \overrightarrow{D}_m^{Iter,\alpha} \\
\overrightarrow{V}_m^{Iter,\beta} &= \overrightarrow{G}_m^{Iter,\beta} + \overrightarrow{H}_m^{Iter,\beta} + \overrightarrow{S}_m^{Iter,\beta} + \overrightarrow{D}_m^{Iter,\beta} \\
\overrightarrow{V}_m^{Iter,\gamma} &= \overrightarrow{G}_m^{Iter,\gamma} + \overrightarrow{H}_m^{Iter,\gamma} + \overrightarrow{S}_m^{Iter,\gamma} + \overrightarrow{I}_m^{Iter,\gamma} + \overrightarrow{D}_m^{Iter,\gamma} + \overrightarrow{R}_m^{Iter,\gamma} \\
\overrightarrow{V}_m^{Iter,\delta} &= \overrightarrow{G}_m^{Iter,\delta} + \overrightarrow{I}_m^{Iter,\delta} + \overrightarrow{R}_m^{Iter,\delta}
\end{aligned} \quad (2)$$

3.1.1 Six general behaviours of horses simulated in HOA

The six general and social behaviours of horses at various ages that form the basis of HOA are detailed below, along with their mathematical implementations.

- Grazing (G): horses are herbivores that graze on grass for 16–20 h each day at all stages of their lives (Krueger and Heinze 2008).
- Hierarchy (H): because horses are not independent, they often follow a leader (Bogner 2011). In a herd of horses, the strongest and most seasoned horse typically leads while the others follow. This is named the hierarchy law, and β and γ horses were demonstrated to follow the law of hierarchy.
- Sociability (S): horses are social animals that can cohabit with other animals. This improves their chances of surviving as well (Krueger and Heinze 2008). This behaviour is seen in horses between the ages of 5 and 15. In HOA, socialisation is defined as a horse's migration toward other horses' positions in the herd.
- Imitation (I): horses learn from one another's good and bad habits and behaviours, by copying them (Bogner 2011). Horses imitate other horses when they are young, and this behaviour continues throughout their lives.
- Defence mechanism (D): to defend themselves, horses engage in fight-or-flight behaviour. The initial reaction of horses is to flee. Moreover, they frequently buck when trapped. Horses also strive to avoid harmful situations with animals such as wolves (Krueger and Heinze 2008; Waring 1983). This behaviour in HOA is characterised as running away from horses that display undesirable behaviours.
- Roaming (R): if horses are not housed in stables, they will graze and travel from one pasture to another in search of food. A horse's grazing location may be abruptly changed. Horses are quite curious creatures since they regularly visit various pastures and attempt to know their surroundings (Waring 1983). In HOA, roaming behaviour is defined as a horse's random mobility within the herd.

Equations (3) to (14) show the matmatical implementation of the six behaviours of horses in HOA:

$$\vec{G}_m^{Iter,AGE} = g_{Iter}(u + \rho\, l) + [X_m^{(Iter-1)}],\ AGE = \alpha, \beta, \gamma, \delta \quad (3)$$

$$g_m^{Iter,AGE} = g_m^{(Iter-1),AGE} \times \omega_g \quad (4)$$

$$\vec{H}_m^{Iter,AGE} = h_m^{Iter,AGE}\left[X_*^{(Iter-1)} - X_m^{(Iter-1)}\right], AGE = \alpha, \beta \text{ and } \gamma \quad (5)$$

$$h_m^{Iter,AGE} = h_m^{(Iter-1),AGE} \times \omega_h \tag{6}$$

$$\vec{S}_m^{Iter,AGE} = s_m^{Iter,AGE}\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right],\ AGE = \beta,\gamma \tag{7}$$

$$S_m^{Iter,AGE} = s_m^{(Iter-1),AGE} \times \omega_s \tag{8}$$

$$\vec{I}_m^{Iter,AGE} = i_m^{Iter,AGE}\left[\left(\frac{1}{pN}\sum_{j=1}^{pN} \hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right],\ AGE = \gamma \tag{9}$$

$$i_m^{Iter,AGE} = i_m^{(Iter-1),AGE} \times \omega_i \tag{10}$$

$$\vec{D}_m^{Iter,AGE} = -d_m^{Iter,AGE}\left[\left(\frac{1}{qN}\sum_{j=1}^{pN} X_j^{(Iter-1)}\right) - X^{(Iter-1)}\right],\ AGE = \alpha,\beta\ and\ \gamma \tag{11}$$

$$d_m^{Iter,AGE} = d_m^{(Iter-1),AGE} \times \omega_d \tag{12}$$

$$\vec{R}_m^{Iter,AGE} = r_m^{Iter,AGE} pX^{(Iter-1)},\ AGE = \gamma\ and\ \delta \tag{13}$$

$$r_m^{Iter,AGE} = r_m^{(Iter-1),AGE} \times \omega_r \tag{14}$$

We can calculate the general velocity by substituting Eqs. (3–14) in Eq. (2). Equations (15–18) can be utilised to obtain the horses' velocity at various ages (δ, γ, β, and α, respectively):

$$\vec{V}_m^{Iter,\delta} = \left[g_m^{(Iter-1),\delta}\omega_g(u+\rho\, l) + [X_m^{(Iter-1)}]\right] + \left[i_m^{(Iter-1),\delta}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right]\right] + \left[r_m^{(Iter-1),\delta}\omega_r pX^{(Iter-1)}\right] \tag{15}$$

$$\begin{aligned}\vec{V}_m^{Iter,\gamma} =& \left[g_m^{(Iter-1),\gamma}\omega_g(\,u+\rho\, l) + [X_m^{(Iter-1)}]\right] + \left[h_m^{(Iter-1),\gamma}\omega_h\left[X_*^{(Iter-1)} - X_m^{(Iter-1)}\right]\right] \\ &+ \left[s_m^{(Iter-1),\gamma}\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right]\right] + \left[i_m^{(Iter-1),\gamma}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right]\right] \\ &- \left[d_m^{(Iter-1),\gamma}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN} X_j^{(Iter-1)}\right) - X^{(Iter-1)}\right]\right] + \left[r_m^{(Iter-1),\gamma}\omega_r pX^{(Iter-1)}\right]\end{aligned} \tag{16}$$

$$\begin{aligned}\vec{V}_m^{Iter,\beta} =& \left[g_m^{(Iter-1),\beta}\omega_g(\,u+\rho\, l) + [X_m^{(Iter-1)}]\right] + \left[h_m^{(Iter-1),\beta}\omega_h\left[X_*^{(Iter-1)} - X_m^{(Iter-1)}\right]\right] \\ &+ \left[s_m^{(Iter-1),\beta}\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right]\right] \\ &- \left[d_m^{(Iter-1),\beta}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN} X_j^{(Iter-1)}\right) - X^{(Iter-1)}\right]\right]\end{aligned} \tag{17}$$

$$\begin{aligned}\vec{V}_m^{Iter,\alpha} =& \left[g_m^{(Iter-1),\alpha}\omega_g(u+\rho\, l) + [X_m^{(Iter-1)}]\right] \\ &- \left[d_m^{(Iter-1),\alpha}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{pN} X_j^{(Iter-1)}\right) - X^{(Iter-1)}\right]\right]\end{aligned} \tag{18}$$

Adult horses (α) start a local search with extreme accuracy around the global optima. The horses of the β group start to search for additional nearby locations around the adult horses (α) with the intention of approaching them. But, horses of the γ group do not like approaching the horses of the α group, and they have a great intention of going to new locations and finding new global optima points. Horses in the δ group are ideal candidates for the random search phase because of their distinct behavioural attributes.

In the field of IDS, researchers have applied various techniques such as machine learning algorithms, data mining, and ensemble methods to improve the accuracy of intrusion detection. After dramatic advances in computer processing units and the availability of public datasets, learning models are becoming increasingly popular for a wide variety of tasks, such as machine vision, pattern recognition, and natural language processing. Researchers in the field of cyber security are following this trend toward using learning models for intrusion detection based on the effectiveness of the methods. Most of the previous

methods based on machine learning have the problem of uncertainty and high attack detection rates. The followings are some of HOA's primary strengths and differences compared to state-of-the-art algorithms.

i. It is a quick optimisation algorithm that uses a variety of parameters to create a coherent algorithm that can discover the best solution as quickly as feasible.
ii. The method is reliable and prevents stagnation phenomena. By using a sorting method in a global matrix, it uses an appropriate approach to prevent local optima entrapment.
iii. It has thecapability to resolve both non-complex and complex optimisation issues. Many alternative algorithms are only effective at handling non-complex problems and are unable to address complex optimisation challenges. This algorithm performs exceptionally well in solving complicated problems in high dimensions because of the numerous control factors based on the behaviour of horses at various ages.
iv. It is very efficient in exploration and exploitation, based on the results of seven well-known test functions as benchmarks. A higher level of exploration and exploitation is achieved in HOA as a result of harmonious relationships between the horse movements.
v. Its components are well-balanced, increasing the efficiency of computational complexity.
vi. It has the capacity to resolve problems in high-dimensional domains with a large number of unknowable variables.
vii. The proposed method can handle a wide range of discrete and multi-objective optimisation problems because it has been transformed into a binary, quantum-inspired, and multi-objective algorithm.

More details about HOA can be found in MiarNaeimi et al. (2021).

3.2 Binary HOA

It is quite easy to develop the binary version of HOA. Before the algorithm starts running, the variables' minimum and maximum values must be specified to be between zero and one. The floor function is used to round the values to vectors of zero and one, before submitting them to the cost function. In this case, the variables are continuous and will remain such; the floor function will only convert them to binary before they enter the cost function. In other words, the cost function operates in a discrete manner while the algorithm attempts to solve a continuous problem. Accordingly, a function establishes the communication between the continuous algorithm and the binary cost function. This operation is carried out using the floor function shown in Eq. (19), where x is a real variable between consecutive integers, m and n, and k is an integer produced when the floor function is applied to the value of x. A similar method of discretising the HOA algorithm has already been used by Hosseinalipour and Ghanbarzadeh (2022a, b), and this can fix the algorithm's continuity issue in solving discrete problems.

$$k = \lfloor x \rfloor \tag{19}$$

3.3 Quantum computing

The smallest piece of information that can be stored in a quantum computer with two states is called a Q-bit in quantum computing. Equation (20) represents the smallest amount of information that could be stored in the Q-bit (Srikanth et al. 2018);

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} : \forall \alpha, \beta \epsilon [0, 1] \tag{20}$$

α and β in Eq. (20) are relevant to the presence of Q-bits in state 0 or state 1 through Eq. (21).

$$|\alpha|^2 + |\beta|^2 = 1 \tag{21}$$

where $|\alpha|^2$ presents the probability of driving the Q-bit into the state 0, and similarly, $|\beta|^2$ presents the probability of driving the Q-bit into state 1. The probability that the Q-bit will end up in the state $|0\rangle$ increases and decreases with the value of $|\alpha|^2$. The greater the value of $|\alpha|^2$, the more likely the Q-bit will end up in the state $|0\rangle$, and vice versa. It is the same for $|\beta|^2$. Therefore, the Q-bit follows the principle of linear superposition of quantum computation according to Eq. (22):

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{22}$$

Equation (23) gives the Q-bits' string represented as a Q-bit individual:

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \alpha_3 & .....\alpha_n \\ \beta_1 & \beta_2 & \beta_3 & .....\beta_n \end{array}\right] \tag{23}$$

n in Eq. (23) is the number of Q-bits in a Q-bit individual, and every bit fulfils Eq. (24).

$$|\alpha_i|^2 + |\beta_i|^2 = 1 : \forall \quad i\epsilon\{1, 2, 3, \ldots, n\} \tag{24}$$

Whenever α and β converge to a value $1/\sqrt{2}$, the probability that a Q-bit will achieve either 0 or 1 states is equal.

In this situation, the linear superposition principle yields Eq. (25) for every possible state with equal probability $(1/\sqrt{2})$.

$$|\varphi\rangle = \sum_{k=1}^{2^n} \frac{1}{\sqrt{2^n}} |X_k\rangle \tag{25}$$

In Eq. (25), X_k signifies the state k as represented by a binary string $(x_1, x_2, \ldots, x_N)$ in which each bit has a binary value of "0" or "1". In every iteration (t), each Q-bit is updated through the use of the Q-gate variation operator $U\left(\Delta\theta_{ji}^{t}\right)$ as given by Eq. (26):

$$\begin{bmatrix} \alpha_{ji}^{t} \\ \beta_{ji}^{t} \end{bmatrix} = U\left(\Delta\theta_{ji}^{t}\right) \begin{bmatrix} \alpha_{ji}^{t-1} \\ \beta_{ji}^{t-1} \end{bmatrix} \tag{26}$$

$U\left(\Delta\theta_{ji}^{t}\right)$ in Eq. 26 is calculated according to Eq. (27).

$$U\left(\Delta\theta_{ji}^{t}\right) = \begin{bmatrix} \cos\left(\Delta\theta_{ji}^{t}\right) & -\sin\left(\Delta\theta_{ji}^{t}\right) \\ \sin\left(\Delta\theta_{ji}^{t}\right) & \cos\left(\Delta\theta_{ji}^{t}\right) \end{bmatrix} \quad \forall j \in n, i \in m \tag{27}$$

In Eq. (27), n is the number Q-bits, and m is the number of Q-bit individuals. $\Delta\theta_{ji}^{t}$ represents the rotation angle of the *jth* Q-bit's *ith* Q-individual for iteration t. $\Delta\theta_{ji}^{t}$ decides the Q-bit's direction and magnitude of variation, for each iteration. According to Eq. (27), for the best solution (B), the $\Delta\theta_{ji}^{t}$ can be predefined based on the search table shown in Table 1 (Srikanth et al. 2018). In Table 1, x_{ji}^{t} is the *ith* individual's *jth* bit in the iteration/generation t for the binary solution X. b_j^t represents the best solution B's *jth* bit for the *tth* iteration. The angle vector $\Theta = [\theta_1, \theta_2, \theta_3, \ldots, \theta_8]^T$ is initialised with the values of $\theta1 = 0$, $\theta2 = 0.01\ \pi$, $\theta3 = -0.01\ \pi$, $\theta4 = 0$, $\theta5 = 0$, $\theta6 = 0$, $\theta7 = 0$, and $\theta8 = 0$ according to the results of the knapsack problems' experimental testing (Narayanan 1999).

3.4 Multi-objective HOA

"Single-objective" optimisers are models that use only one objective function to solve optimisation problems (Chang 2014). Discovering the best solution, which is typically unique, is the goal of a single-objective optimisation problem (Coello and Lamont 2004). However, many real-world and engineering challenges have multiple objective functions, which are considered multi-objective optimisation challenges. Therefore, there is a need for multi-objective models to address those problems. Although, in many instances, a multi-objective optimisation problem's defined objective functions are incompatible (Chang 2014), and the goals are also not compatible (Mirjalili 2016).

The problem of NID is multi-objective, and the two objectives pursued in the feature selection of this problem would be the number of features and classification accuracy. The number of features used in this problem should be kept to a minimum, and the classification accuracy should be as high as possible. The higher classification accuracy means that after the classification process is completed, the majority of the intrusions are classified into their correct category (meaning that the intrusions are detected), and the number of incorrect categories is minimal. Also, due to the fact that the classification is carried out with the proposed algorithm's selected features, the multi-objective HOA algorithm, the number of features should be minimised to avoid the method's complexity. Since more than one objective function should be considered in this problem, it is important to use multi-objective optimisation approaches. The main characteristic of such techniques is that they offer multiple candidate solutions. Each of these solutions explains how multiple goal functions are balanced (Khanmohammadi et al. 2021; Khodadadi et al. 2021). The multi-objective version of HOA has been used in previous studies for different applications (Hosseinalipour and Ghanbarzadeh 2022a, b), and a similar technique is adopted in the current study to convert HOA to a multi-objective algorithm.

Table 1 Search table of sample rotation angles (T = True, F = False) adapted from Srikanth et al. (2018)

x_{ji}^{t}	b_j^t	F $(x_j^t) \le$ F (B^t)	$\Delta\theta_{ji}^{t}$
0	0	F	θ_1
0	1		θ_2
1	0		θ_3
1	1		θ_4
0	0	T	θ_5
0	1		θ_6
1	0		θ_7
1	1		θ_8

4 Proposed approach

NIDs are network nodes that attempt to identify intrusive activities and prevent unauthorised intrusions in a computer network as much as feasible. The NID system is considered a node in the network that can be located in different locations and has a monitoring functionality. Each NID system can serve independently as a monitoring agent and is responsible for recognising intrusions into local nodes or clusters. They identify intrusions that impact all of the network or a part of it. Conceptually, a NID node consists of six parts: a 'data collection module', a 'local detection engine', a 'cooperative detection engine', a 'local response module', a 'global response module', and a 'secure communication module' that provides a highly reliable communication channel in between IDS agents (Zhang and Lee 2005). Various methods

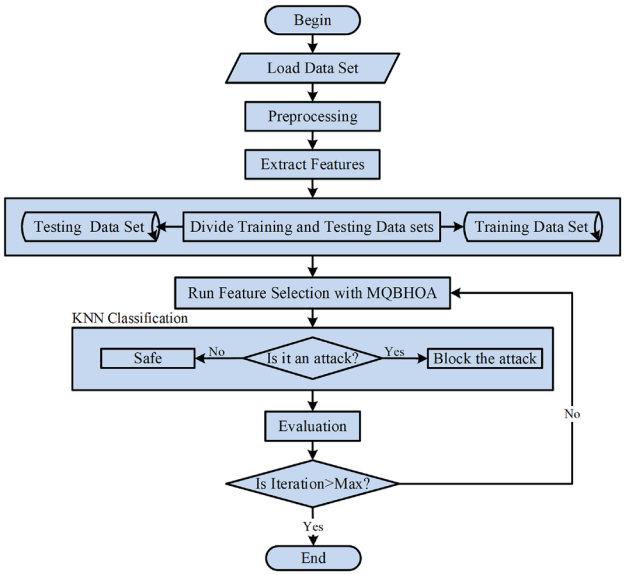

Fig. 2 Flowchart of the proposed approach

have been used for local intrusion detection, and the network's intrusion detection node functions differently. We have also used a different approach in this article. Figure 2 depicts a flowchart of the suggested approach's main steps.

As can be seen in the flowchart in Fig. 2, in the beginning, we enter the data set into the system. In the next stage, pre-processing is performed on the data set. Some features related to one or more samples may be missing due to noise or other issues and no longer have valid values. In this step, a proper solution must be used to determine the values of the missing data to process data sets that have missing data. NSL-KDD and CSE-CIC-IDS2018 are normal dataset and does not require a pre-processing step. In the next stage, features are extracted. Feature extraction is one of the important steps in machine learning, and extracting the desired features improves classification accuracy. Feature extraction methods are widely used in classification. The feature extraction phase was already carried out initially on the dataset sample, and the extracted features are listed in Table 2, along with their descriptions.

The next step divides the dataset into two data sets: the training data set and the testing data set. In this way, 80% of the data is applied for training the data set, and the remaining 20% is used for testing. The feature selection in the proposed method is carried out on the training part of the data set.

In the next step, the main part of the algorithm is executed, which is the feature selection or the process of reducing the number of input data to decrement the computational cost using the proposed method. Next, data classification is carried out according to the selected features to classify data into two classes, safe and intrusion, and then the classified data is evaluated. Finally, if the algorithm termination conditions are met and all data is classified, the algorithm is terminated. Otherwise, it returns to the feature selection stage to continue the steps until the algorithm ends.

4.1 Quantum-inspired HOA

One of the problems which could be solved in a shorter time using quantum-inspired algorithms compared to classical algorithms is search problems. The single form of a search query can be expressed as follows. $S := \{x_1, x_2, \ldots, x_N\}$ is a set that contains *N* objects. A function such as $f : S \rightarrow \{0, 1\}$ is defined on the set S. We know that the value of the function f is only zero on one of the elements of this set, which we denote by W, and on the other elements of the set S, the value of this function is equal to one. W is one of the x_i; however, we do not know which one it is. In the absence of any additional information, all we need to do is to give different x_i one by one to the function, and check the output of the f function. Whenever the output equals one, it means that the element given to the function was W. On average, we need to check the output of the function $O(\frac{N}{2})$ times to be able to access W. However, a creative application of the quantum superposition and parallelism principle can help reduce this value to $(O\sqrt{N})$, which is a significant reduction for large *N*s. Obviously, the class of this problem has not changed with this invention, and it is still in the class of polynomial problems. Still, this improvement is significant because of a search algorithm's role in most other algorithms. In converting the discrete HOA to a quantum-inspired algorithm, the approach used by Srikanth et al. (2018) was followed.

4.1.1 Binary HOA with quantum computing

The position of each horse in the new method is either "0" or "1" based on the probabilities $|\alpha|^2$ and $|\beta|^2$, respectively. The updated Q-bit is replaced with HOA's updating process. Thus, in quantum-inspired binary HOA, the update coefficients are replaced with a Q-gate change operator. Using $|\beta_i|^2$, which is a probability that shifts the mode to "1", the position of the horses is updated.

4.2 Network intrusion detection using multi-objective quantum-inspired binary HOA

The new NID method applied two data sets on which the initial processing has already been carried out, and a series of features have been extracted. The method chooses a number of extracted features that detect network intrusion.

Table 2 NSL-KDD dataset features

Feature name		Type	Description
1	Duration	Continuous	Length (number of seconds) of the connection
2	Protocol type	Discrete	Type of the protocol, e.g., tcp, udp, etc
3	Service	Discrete	Network service on the destination, e.g., http, telnet, etc
4	src_bytes	Continuous	Number of data bytes from source to destination
5	dst_bytes	Continuous	Number of data bytes from destination to source
6	Flag	Discrete	Normal or error status of the connection
7	Land	Discrete	1 if the connection is from/to the same host/port; 0 otherwise
8	Wrong_fragment	Continuous	Number of "wrong" fragments
9	Urgent	Continuous	Number of urgent packets
10	Hot	Continuous	Number of "hot" indicators
11	Num_failed_logins	Continuous	Number of failed login attempts
12	Logged_in	Discrete	1 if successfully logged in; 0 otherwise
13	Num_compromised	Continuous	Number of "compromised" conditions
14	Root_shell	Discrete	1 if root shell is obtained; 0 otherwise
15	su_attempted	Discrete	1 if "su root" command is attempted; 0 otherwise
16	Num_root	Continuous	Number of "root" accesses
17	Num_file	Creations	Continuous number of file creation operations
18	Num_shells	Continuous	Number of shell prompts
19	Num_access_files	Continuous	Number of operations on access control files
20	Num_outbound_cmds	Continuous	Number of outbound commands in an ftp session
21	Is_hot_login	Discrete	1 if the login belongs to the "hot" list; 0 otherwise
22	Is_guest_login		1 if the login is a "guest" login; 0 otherwise
23	Count	Continuous	Number of connections to the same host as the current connection in the past two seconds
24	Serror_rate	Continuous	% of connections that have "SYN" errors
25	Rerror_rate	Continuous	% of connections that have "REJ" errors
26	Same_srv_rate	Continuous	% of connections to the same service
27	Diff_srv_rate	Continuous	% of connections to different services
28	srv_count	Continuous	Number of connections to the same service as the current connection in the past two seconds
29	srv_serror_rate	Continuous	% of connections that have "SYN" errors
30	srv_rerror_rate	Continuous	% of connections that have "REJ" errors
31	srv_diff_host_rate	Continuous	% of connections to different hosts
32	dst_host_count	Continuous	count for the destination host
33	dst_host_srv_count	Continuous	srv_count for the destination host
34	dst_host_same_srv_rate	Continuous	Same_srv_rate for the destination host
35	dst_host_diff_srv_rate	Continuous	Diff_srv_rate for the destination host
36	dst_host_same_src_port_rate	Continuous	Same_src_port_rate for the destination host
37	dst_host_diff_host_rate	Continuous	Diff_host_rate for the destination host
38	dst_host_serror_rate	Continuous	Serror_rate for the destination host
39	dst_host_srv_serror_rate	Continuous	srv_serror_rate for the destination host
40	dst_host_rerror_rate	Continuous	Rerror_rate for the destination host
41	dst_host_srv_rerror_rate	Continuous	srv_serror_rate for destination hos"

This is accomplished using the natural processes of the HOA algorithm. The process of selecting features has four steps: feature subsets generation, subsets evaluation, termination criteria review, and results validation (Kumar et al. 2016). The feature subset is generated in the data set. In this subset, candidate features are searched based on the search strategy of multi-objective quantum-inspired binary HOA (MQBHOA). Afterwards, the candidate subsets are assessed and contrasted with the evaluation feature's best value from the previous iteration. The previous best

Table 3 Simulation parameters and their values

Simulation parameter	Value
Population_size	50
Problem_size	42
Number of packets in train data set	25192
Binary number of packets in test data set	4507
Max_generations	10
Test data set	22544
Train data set	125973

subset is used unless a better one is generated. Up until the MQBHOA termination requirement is met, this subset generation and evaluation process is repeated. MQBHOA is iterated several times until the ideal global solution is found. The classifier's accuracy for the candidate subset at the end of each cycle is calculate by the fitness function. The process of generating candidates, determining their fitness, and evaluating them doesn't stop until the termination requirements are satisfied. Generally, termination criteria are defined based on two factors: the error rate and the total number of iterations. The algorithm terminates if either the error rate falls below a predetermined level or the algorithm exceeds through too many iterations.

5 Simulation and evaluation

MATLAB R2014a on a Microsoft Windows PC with a 64-bit core i5 CPU and 4G memory is used for the simulation of the new algorithm. The new algorithm's performance was evaluated on two different data sets, NSL-KDD and CSE-CIC-IDS2018, for detecting computer network intrusions. The evaluation results on both data sets are discussed in the following sub-sections.

5.1 Performance evaluation of MQBHOA on NSL-KDD data set

The NSL-KDD data set is the recent version of the KDD99 data set. This data set contains 125,973 training records and 22,544 testing records. Therefore, 80% of the data is used for training (file name: *KDDTrain_20.arff*), and the remaining 20% is used for testing (file name: *KDDTest-21.arff*). Each record of this data set is either a normal record or belongs to one of the 22 categories of attacks. All attacks have been categorised into four major categories: DoS, U2R, R2L, and Probing, and the data set has 41 features. Table 2 shows the name, type and description of all 41 features.

The parameters used in the simulation of the algorithm and the value of each parameter are demonstrated in Table 3.

The average accuracy of the new method for separating normal samples from abnormal samples is between 96 and 99%. This percentage of accuracy is valid in the case of a two-class training mode for the proposed approach (when the algorithm is trained to identify whether there is an attack or not- intrusion or no intrusion). Table 4 and Fig. 3 show the accuracy rate of the introduced method in intrusion detection compared with other methods such as Naïve Bayes, Self Organisation Map (SOM) (Ibrahim et al. 2013), RF, SVM, and Multinomial Naïve Bayes (+N2B) (Panda et al. 2010) methods using the NSL-KDD data set. The results are demonstrated in 10, 50 and 100 iterations of the algorithms.

As shown in Table 4 and Fig. 3, the new intrusion detection method has the highest accuracy rate compared to several classical and recent methods. This is because the proposed method with a suitable initialisation, useful fitness

Fig. 3 Comparison of the new method's accuracy rate with other similar methods

Table 4 Comparison of the accuracy of intrusion detection of various methods

Method	10 Iterations	50 Iterations	100 Iterations
Naïve bayes	59.75	63.24	65.01
Self organisation map (SOM)	53.54	56.05	59.08
RF	70.87	70.58	72.23
SVM	58.21	61.34	62.43
Multinomial naïve bayes (+N2B)	92.65	95.12	97.37
MQBHOA	96.41	98.76	99.36

Table 5 Comparison of GWO, WOA, FFA, MFO, PSO, and KNN with MQBHOA in terms of accuracy of classification

Method	10 Iteration	50 Iteration	100 Iteration
GWO	84	90	99
WOA	86	92	98
FFA	83	87	98
MFO	80	86	96
PSO	77	85	94
KNN	65	73	80
MQBHOA	92	97	99.8

function and accurate evaluation function, well escapes from the local optima in the problem state space and moves towards the global optima solution; thus, it is more successful in identifying normal traffics or attack states in the network.

Next, we examined and compared MQBHOA with GWO, WOA, FFA, MFO, PSO, and KNN regarding the classification accuracy of different types of intrusions in networks. In this simulation, the population size is set to 20, and the iteration numbers are 10, 50, and 100. The results of this simulation are presented in Table 5 and Fig. 4.

According to Table 5 and Fig. 4, the proposed approach has obtained substantially favourable results in the classification accuracy of the intrusion detection problem compared to GWO, WOA, FFA, MFO, PSO and KNN as the number of iterations increased. In the first iterations, MQBHOA performed comparably to some of the other methods; however, it has been able to surpass them with the increase in the iteration number due to the adoption of the quantum-inspired technique in generating solutions that are in the opposite search space.

Next, in terms of the F-Measure criterion of the proposed algorithm, we examined GWO, WOA, FFA, MFO, PSO and KNN. The results of the simulation are shown in Table 6 and Fig. 5. The population size in this simulation is set to 20, and the parameters for the quantity of iterations are 10, 50, and 100.

The test results, according to Table 6 and Fig. 5, show that MQBHOA, by increasing the number of iterations, has obtained much better results and accuracy compared to the GWO, WOA, FFA, MFO, PSO and KNN algorithms in the classifying the types of intrusions. In the first iterations,

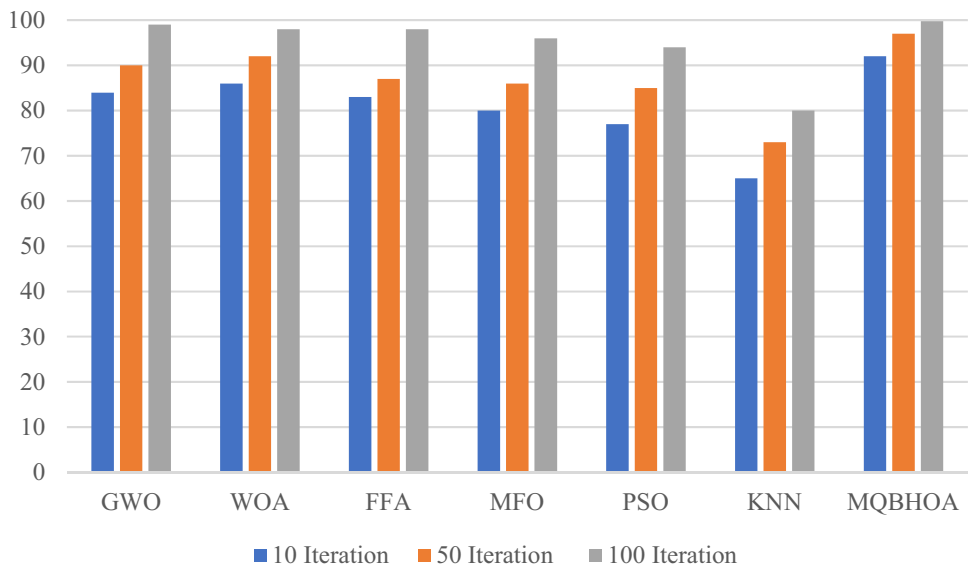

Fig. 4 Comparison of the performance of GWO, WOA, FFA, MFO, PSO, and KNN with MQBHOA in terms of classification accuracy for intrusion detection

Table 6 Comparison of GWO, WOA, FFA, MFO, PSO and KNN with MQBHOA in various iterations in terms of the F-measure of intrusions in the network

Method	10 Iteration	50 Iteration	100 Iteration
GWO	81	90	99
WOA	82	89	98
FFA	80	86	95
MFO	78	85	95
PSO	76	82	93
KNN	68	72	91
MQBHOA	94	98	99

MQBHOA performed similarly to other algorithms, but with the increase in the iterations, as a result of quantum-based methods to create solutions in the opposite search space, it has achieved its superiority over the GWO, WOA, FFA, MFO, PSO, KNN.

MQBHOA is also evaluated regarding the accuracy, precision, sensitivity, and F1-score of the NID, and it has been compared with KNN-GWO, KNN, MLP, SVM and NB classifiers. Table 7 and Fig. 6 show the result of this evaluation. The opposition-based, binary and multi-objective version of HOA has already been introduced and used in another study for a different application, MOBHOA (Hosseinalipour

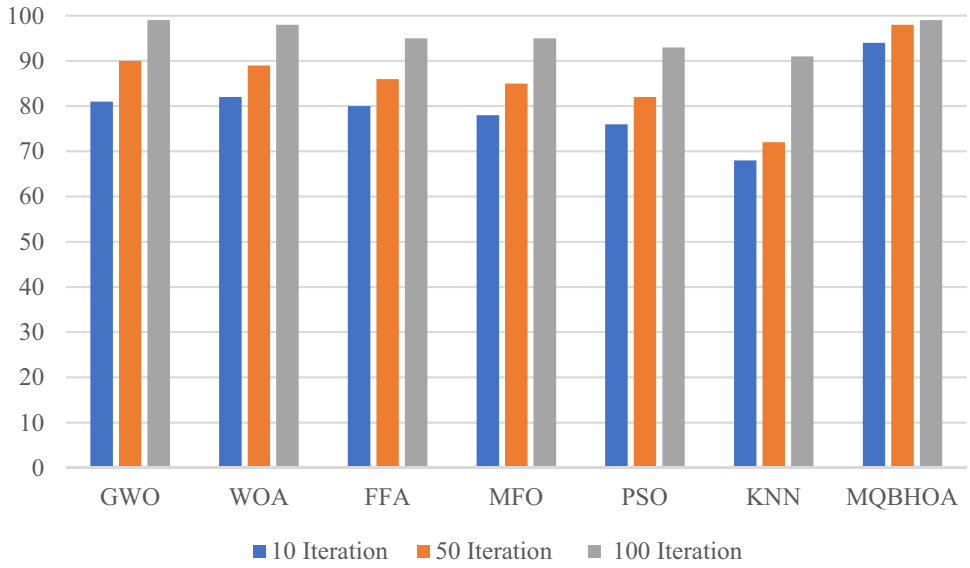

Fig. 5 Performance comparison of GWO, WOA, FFA, MFO, PSO, and KNN with MQBHOA in terms of F-Measure

Table 7 Evaluation comparison of MQBHOA with other algorithms regarding precision, sensitivity, accuracy and F1-score (NSL-KDD data set)

	KNN	NB	SVM	MLP	KNN- GWO	KNN- MOB-HOA	KNN-MQB-HOA
Precision	74	59	79	75	93	98	99
Accuracy	71	58	81	72	92	97	99
Sensitivity	72	63	80	70	89	97	98
F1-score	68	60	78	74	88	97	97

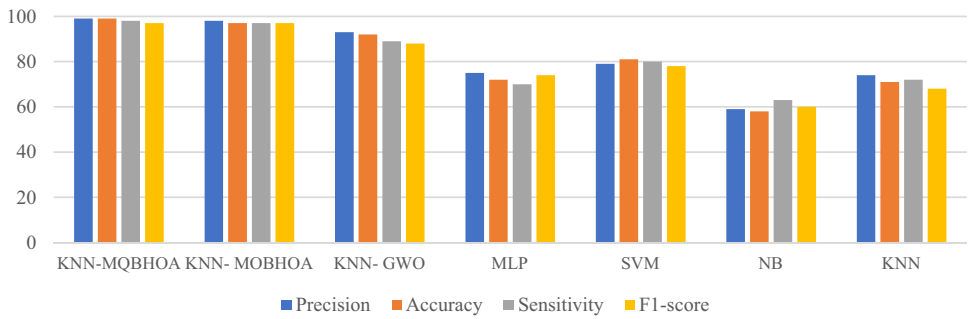

Fig. 6 Evaluation comparison of the MQBHOA regarding precision, sensitivity, accuracy and F1-score in intrusion detection

Table 8 The results of the confusion matrix

		Prediction	
		Yes	No
Actual	Yes	22499	7
	No	9	29

and Ghanbarzadeh 2022a). The evaluation results of the application of MOBHOA for intrusion detection are also included here for comparison purpose, and as can be seen from Tables 7 and 9, MQBHOA demonstrates better performance for NID in both data sets, compared to MOBHOA.

The purpose of the new approach in Table 7 is feature selection, and as mentioned earlier, the NID process is completed in two main stages, the first stage is feature selection, and the second stage is classification. The results shown in Table 7 are based on the fact that the feature selection phase of the new method is carried out with the MQBHOA, and its classification step is performed by KNN. Selecting the optimal feature in MQBHOA has increased accuracy, precision and sensitivity and reduced execution time. The rationale behind this is that operations are performed on significant features rather than redundant or insignificant features when selecting the best feature. Therefore, the execution time of the algorithm is also reduced. In this experiment, we kept the KNN constant in all feature selections and ran the execution, which is obvious in the results shown in Table 7. In order to demonstrate the superiority of the new algorithm, the results of the confusion matrix for 22544 tested data are provided in Table 8.

5.2 Performance evaluation of MQBHOA on CSE-CIC-IDS2018 data set

Due to the changing behaviour patterns of network attacks, we evaluated the proposed algorithm's performance on the CSE-CIC-IDS2018 data set as well, which is a new data set and improved in the dynamic environment for evaluation (Sharafaldin 2017). This data set has several critical features for IDS, such as the fact that the variety of attacks and the anonymity of existing protocols are used in this data set. These datasets offer a conceptual understanding of numerous application models, network devices, and protocols in addition to detailed knowledge of the attacks carried out. This dataset was formed in 2018 and had 18 classes and 80 features Panigrahi and Borah (2018). Table 9 and Fig. 7 show the intrusion detection accuracy of the new method using the CSE-CIC-IDS2018 dataset.

The evaluation results in Table 9 regarding the application of KNN-MQBHOA in NID compared with other classifiers, e.g. NB, SOM, RF, SVM, MNB, and KNN-MOBHOA, indicate that the new approach has improved significantly in precision, accuracy, sensitivity and F1-score.

According to the evaluation results, compared to other basic metaheuristic algorithms, the average size of feature selection and classification accuracy are both improved by MQBHOA when executed on the mentioned data set. The findings also show that it performs substantially better than other similar algorithms in NID.

Table 9 Evaluation comparison of MQBHOA with other algorithms regarding precision, sensitivity, accuracy and F1-score (CSE-CIC-IDS2018 data set)

	NB	SOM	RF	SVM	Multinomial naïve bayes (+N2B)	KNN- MOBHOA	KNN-MQBHOA
Precision	94.21	92.98	94.14	95.77	96.46	98.41	99.56
Accuracy	92.65	91.44	93.44	95.25	95.45	99.27	99.78
Sensitivity	95.1	95.01	96.17	96.85	97.02	97.95	99.87
F1-score	93.41	92.21	94.28	95.47	96.49	97.57	99.65

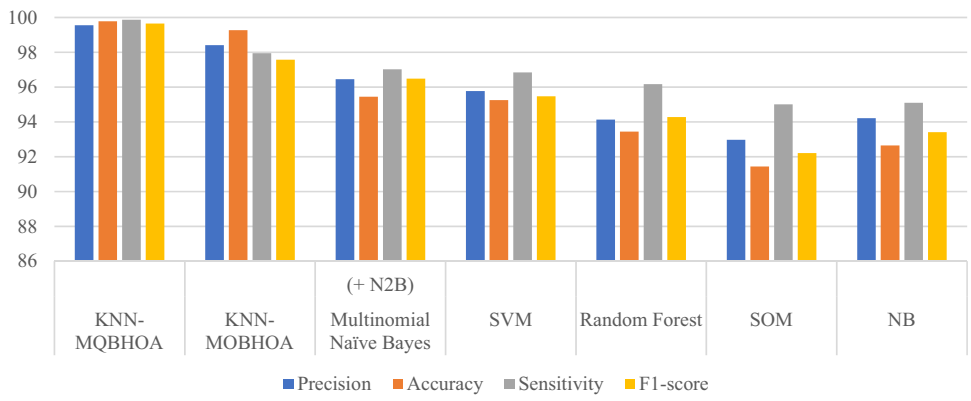

Fig. 7 Evaluation comparison of MQBHOA with other algorithms regarding precision, sensitivity, accuracy and F1-score (CSE-CIC-IDS2018 data set)

Table 10 Number of features selected by the MQBHOA and compared algorithms in the feature selection phase

Algorithm	Intrusion type	#Feature	Selected features
Naïve bayes	Normal	7	1, 3, 4, 5, 6, 13, 17
	DOS	5	2, 3, 5, 33, 38
	Probe	17	1, 5, 9, 11, 13, 14, 17, 18, 19, 22, 24, 25, 28, 30, 31, 32, 39
	R2L	11	3, 4, 6, 9, 11, 22, 25, 33, 38, 39, 40
	U2R	18	1, 3, 5, 6, 7, 8, 9, 15, 16, 17, 15, 22, 26, 29, 31, 32, 33, 39
SOM	Normal	6	3, 5, 6, 21, 23, 34
	DOS	8	3, 5, 6, 23, 24, 35, 36, 41
	Probe	12	3, 5, 12, 23, 24, 27, 28, 32, 33, 35, 38, 40,
	R2L	14	1, 3, 5, 6, 10, 13, 22, 2, 32, 33, 35, 37, 39, 41
	U2R	10	1, 2, 3, 6, 14, 23, 24, 32, 34, 39
Multinomial naïve bayes (+N2B)	Normal	11	2, 10, 12, 23, 26, 27, 33, 34, 35, 39, 40
	DOS	23	2, 3, 5, 6, 9, 10, 11, 14, 16, 17, 18, 19, 24, 25, 26, 28, 31, 33, 32, 36, 39, 40, 41
	Probe	8	12, 27, 28, 33, 34, 35, 40, 41
	R2L	5	3, 10, 22, 38, 40
	U2R	5	2, 13, 14, 17, 32
SVM	Normal	9	1, 2, 3, 5, 6, 12.26, 33, 34, 39
	DOS	15	1, 3, 5, 10, 11, 17, 16, 18, 19, 24, 28, 31, 35, 37, 39
	Probe	11	2, 3, 5, 6, 9, 12, 24, 28, 35, 33, 41
	R2L	10	1, 3, 5, 6, 10, 23, 24, 33, 37, 39
	U2R	9	1, 2, 3, 4, 5, 6, 14, 32, 33
RF	Normal	10	1, 2, 3, 5, 6, 10, 12, 26, 39, 41
	DOS	18	1, 2, 4, 5, 6, 9, 11, 14, 17, 18, 19, 22, 25, 29, 31, 33, 36, 39
	Probe	7	12, 13, 2628, 33, 35, 40, 41
	R2L	9	1, 3, 5, 15, 22, 24, 33, 37, 41
	U2R	9	1, 2, 3, 5, 13, 21, 23, 33, 39
MQBHOA	Normal	5	1, 3, 5, 6, 34
	DOS	4	3, 5, 23, 38
	Probe	8	2, 3, 5, 6, 9, 13, 26, 32
	R2L	8	3, 5, 6, 10, 15, 22.29, 32, 33
	U2R	8	1, 2, 3, 5, 6, 14, 21, 33

Table 10 demonstrates the selected features during the feature selection process in the evaluation stage, using MQBHOA and the other compared algorithms. This table shows the type of intrusion, important features in each type of intrusion, and the number of features and selected features. Table 10 clearly demonstrates that the number of selected features by the proposed solution is less than the compared algorithms, indicating the better performance of MQBHOA at the feature selection stage.

6 Limitations and future work

The current study's findings demonstrate outstanding results in the performance of the HOA algorithm in addressing the network intrusion detection problem. However, some limitations are associated with the algorithm's application in solving this problem.

Firstly, the complexity of intrusion detection systems and the diversity of attacks they need to detect, make it challenging to develop a general-purpose optimisation algorithm that can be applied to addressing all intrusion detection problems. This means that the HOA algorithm may need to be adapted or modified to suit the specific requirements of each intrusion detection problem, which can be time-consuming.

Secondly, most of the techniques used in the literature, including the proposed method, are examined on the basis of a limited number of supervised data sets, as they need a set of labelled data to build supervised learning models; however, obtaining labelled data is usually difficult and expensive, and large amounts of raw and unlabelled data could increase the cost and time of data analysis. Therefore, the application of semi-supervised methods using metaheuristic algorithms can considerably reduce these costs. The use of semi-supervised approaches also has two other advantages: the tests are conducted on real data, and the new method is capable of

recognising new types of unknown or malicious traffic. Therefore, future research can consider employing novel semi-supervised techniques in addressing network intrusion detection.

Despite the above-mentioned limitations, the HOA algorithm has been demonstrated to be effective in solving a range of optimisation problems, and its application to intrusion detection holds promise for improving the accuracy and efficiency of NID systems. There are several directions that future research can take in using HOA for solving intrusion detection problems.

The HOA algorithm can be integrated with other machine learning methods, such as decision trees, neural networks, or SVMs, to improve its performance in detecting intrusion. It can also be adapted to detect specific types of intrusions, such as insider attacks, denial-of-service attacks, or data exfiltration, by modifying the objective function and constraints to reflect the characteristics of each type of intrusion. This algorithm can also be extended to dynamically adapt to changes in the network environment, such as new attacks or changes in network traffic patterns, by incorporating real-time feedback and including it in the optimisation process.

HOA can be optimised to handle large-scale networks by developing more efficient techniques for updating the positions of horses and reducing the computational overhead of the optimisation process. This algorithm can also be evaluated on a larger set of benchmark intrusion detection datasets, and its performance can be compared to future optimisation algorithms and machine learning techniques to further determine its strengths and weaknesses in addressing this problem.

Future research can also consider the application of various metaheuristics as well as semi-supervised algorithms in addressing the intrusion detection problem to achieve higher accuracy and lower computational complexity in detecting a wider range of attacks and intrusions in computer networks.

7 Conclusion

Intrusion detection is essentially considered a problem of classification, and one of the processes in classification is feature selection. For large-scale data, feature selection reduces detection time and cost and improves classification efficiency. In this paper, a new method, MQBHOA, was introduced for intrusion detection in computer networks. This approach employs the HOA metaheuristic optimisation algorithm, which is a nature-inspired and robust algorithm and has a great performance in addressing problems in high dimensions. In order to use HOA for network intrusion detection, first, it was discretised, then transformed to quantum-inspired, and finally, converted to multi-objective. Then the resulting algorithm was used to detect and classify different types of intrusions in computer networks. Two different data sets, NSL-KDD and CSE-CIC-IDS2018, were used in evaluating the new algorithm's performance, and it was compared to various algorithms such as GWO, WOA, FFA, MFO, PSO, KNN, RF, NB, SOM, SVN, and MNB(+N2B) in feature selection and classification. The evaluation and comparison results indicate that the MQBHOA outperforms other approaches.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability The data and code that support the current study's findings are available from the corresponding author on reasonable request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abraham JA, Bindu V (2021) intrusion detection and prevention in networks using machine learning and deep learning approaches: a review. 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)

Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F (2021) Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol 32(1):e4150

Almomani O (2021) A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system. Comput Mater Contin 68(1):409–429

Basu S, Basu M (2021) Horse herd optimization algorithm for fuel constrained day-ahead scheduling of isolated nanogrid. Appl Artif Intell 35(15):1250–1270

Basu S, Kumar S, Basu M (2022) Horse herd optimization algorithm for economic dispatch problems. Eng Optim. https://doi.org/10.1080/0305215X.2022.20353781-17

Bataghva M (2017) Efficiency and accuracy enhancement of intrusion detection system using feature selection and cross-layer mechanism. The University of Western Ontario, Canada

Binbusayyis A, Vaiyapuri T (2021) Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. Appl Intell 51(10):7094–7108

Bogner F (2011) A comprehensive summary of the scientific literature on Horse Assisted Education in Germany. Van Hall Larenstein, Leeuwarden

Breiman L (2001) Random forests. Statistics Department. University of California, Berkeley, p 4720
Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
Chang K-H (2014) Design theory and methods using CAD/CAE: The computer aided engineering design series. Academic Press, Cambridge
Coello CAC, Lamont GB (2004) Applications of multi-objective evolutionary algorithms, vol 1. World Scientific, Singapore
Deore B, Bhosale S (2022) Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection. IEEE Access 10:65611–65622
Duda RO, Hart PE, Stork DG (1995) Pattern classification and scene analysis, 2nd edn. Wiley Interscience, Hoboken, pp 13–14
Dwivedi S, Vardhan M, Tripathi S (2021) Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection. Clust Comput. https://doi.org/10.1007/s10586-020-03229-5
Elmanakhly DA, Saleh M, Rashed EA, Abdel-Basset M (2022) BinHOA: Efficient binary horse herd optimization method for feature selection: analysis and validations. IEEE Access 10:26795–26816
Evangeline SI, Rathika P (2022) Wind farm incorporated optimal power flow solutions through multi-objective horse herd optimization with a novel constraint handling technique. Expert Syst Appl 194:116544
Hammad M, El-medany W, Ismail Y (2020) Intrusion detection system using feature selection with clustering and classification machine learning algorithms on the unsw-nb15 dataset. 2020 international conference on innovation and intelligence for informatics, computing and technologies (3ICT)
Hosseinalipour A, Ghanbarzadeh R (2022a) A novel approach for spam detection using horse herd optimization algorithm. Neural Comput Appl 34(15):13091–13105
Hosseinalipour A, Ghanbarzadeh R (2022b) A novel metaheuristic optimisation approach for text sentiment analysis. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-022-01670-z
Ibrahim LM, Basheer DT, Mahmod MS (2013) A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network. J Eng Sci Technol 8(1):107–119
Jiang H, He Z, Ye G, Zhang H (2020a) Network intrusion detection based on PSO-XGBoost model. IEEE Access 8:58392–58401
Jiang K, Wang W, Wang A, Wu H (2020b) Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access 8:32464–32476
Karatas G, Demir O, Sahingoz OK (2020) Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. IEEE Access 8:32150–32162
Karim NSA, Albuolayan A, Saba T, Rehman A (2016) The practice of secure software development in SDLC: an investigation through existing model and a case study. Secur Commun Netw 9(18):5333–5345
Khanmohammadi S, Kizilkan O, Musharavati F (2021) Multiobjective optimization of a geothermal power plant. Thermodynamic analysis and optimization of geothermal power plants. Elsevier, Amsterdam, pp 279–291
Khodadadi N, Azizi M, Talatahari S, Sareh P (2021) Multi-objective crystal structure algorithm (MOCryStAl): introduction and performance evaluation. IEEE Access 9:117795–117812
Krishnaveni S, Sivamohan S, Sridhar S, Prabakaran S (2021) Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. Clust Comput 24(3):1761–1779
Krueger K, Heinze J (2008) Horse sense: social status of horses (*Equus caballus*) affects their likelihood of copying other horses' behavior. Anim Cogn 11:431–439
Kumar A, Khorwal R, Chaudhary S (2016) A survey on sentiment analysis using swarm intelligence. Indian J Sci Technol 9(39):1–7
Li Y, Ghoreishi S-M, Issakhov A (2022) Improving the accuracy of network intrusion detection system in medical IoT systems through butterfly optimization algorithm. Wireless Pers Commun 126(3):1999–2017
Mehrabi N, Pashaei E (2021) Application of Horse Herd Optimization Algorithm for medical problems. 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)
MiarNaeimi F, Azizyan G, Rashki M (2021) Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems. Knowl-Based Syst 213:106711
Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27:1053–1073
Moghanian S, Saravi FB, Javidi G, Sheybani EO (2020) GOAMLP: Network intrusion detection with multilayer perceptron and grasshopper optimization algorithm. IEEE Access 8:215202–215213
Narayanan A (1999) Quantum computing for beginners. Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)
Nazir A, Khan RA (2021) A novel combinatorial optimization based feature selection method for network intrusion detection. Comput Secur 102:102164
Ozkan-Okay M, Samet R, Aslan Ö, Gupta D (2021) A comprehensive systematic literature review on intrusion detection systems. IEEE Access 9:157727–157760
Panda M, Abraham A, Patra MR (2010) Discriminative multinomial naive bayes for network intrusion detection. 2010 Sixth International Conference on Information Assurance and Security
Panigrahi R, Borah S (2018) A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. Int J Eng Technol 7(3.24):479–482
Pingale SV, Sutar SR (2022) Remora whale optimization-based hybrid deep learning for network intrusion detection using CNN features. Expert Syst Appl 210:118476
Rajeshwari J, Sughasiny M (2022) Skin cancer severity prediction model based on modified deep neural network with horse herd optimization. Opt Mem Neural Netw 31(2):206–222
Sharafaldin G, Sharafaldin I, Gharib A, Lashkari AH, Ghorbani AA (2017) Towards a reliable intrusion detection benchmark dataset. Softw Netw 2017(177200):10.13052
Sharma RK, Issac B, Kalita HK (2019) Intrusion detection and response system inspired by the defense mechanism of plants. IEEE Access 7:52427–52439
Srikanth K, Panwar LK, Panigrahi BK, Herrera-Viedma E, Sangaiah AK, Wang G-G (2018) Meta-heuristic framework: Quantum inspired binary grey wolf optimizer for unit commitment problem. Comput Electr Eng 70:243–260
Su T, Sun H, Zhu J, Wang S, Li Y (2020) BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE Access 8:29575–29585
Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M (2016) Deep learning approach for network intrusion detection in software defined networking. 2016 international conference on wireless networks and mobile communications (WINCOM)
Toldinas J, Venčkauskas A, Damaševičius R, Grigaliūnas Š, Morkevičius N, Baranauskas E (2021) A novel approach for network intrusion detection using multistage deep learning image recognition. Electronics 10(15):1854
Waring GH (1983) Horse behaviour. The behavioral traits and adaptations of domestic and wild horses, including ponies. Noyes Publications, Norwich

Wu P, Guo H (2019) LuNET: a deep neural network for network intrusion detection. 2019 IEEE symposium series on computational intelligence (SSCI)

Wu Z, Wang J, Hu L, Zhang Z, Wu H (2020) A network intrusion detection method based on semantic re-encoding and deep learning. J Netw Comput Appl 164:102688

Yang Y, Zheng K, Wu B, Yang Y, Wang X (2020) Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. IEEE Access 8:42169–42184

Yu Y, Bian N (2020) An intrusion detection method using few-shot learning. IEEE Access 8:49730–49740

Zhang Y, Lee W (2005) Security in mobile ad-hoc networks. Ad hoc networks: technologies and protocols. Springer, Boston, pp 249–268

Zhang X, Chen J, Zhou Y, Han L, Lin J (2019) A multiple-layer representation learning model for network-based attack detection. IEEE Access 7:91992–92008

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.