Expert Systems With Applications 213 (2023) 118745

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning

Orieb Abu Alghanam [a,*], Wesam Almobaideen [a,b], Maha Saadeh [c], Omar Adwan [d,a]

[a] Department of Computer Science, University of Jordan, Amman, Jordan
[b] Department of Electrical Engineering and Computing Sciences, Rochester Institute of Technology, Dubai, United Arab Emirates
[c] Department of Computer Engineering and Informatics, Middlesex University Dubai, Dubai, United Arab Emirates
[d] Department of Computer Science, Al-Ahliyya Amman University, Amman, Jordan

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Ensemble learning
LS-PIO
NIDS
One-class classifiers
PIO

ABSTRACT

With the rapid growth of the number of connected devices that exchange personal, sensitive, and important data through the IoT based global network, attacks that are targeting security services are increasing as well. Accordingly, there is a need for security solutions that are suitable for IoT environment. A network intrusion detection system (NIDS) is a solution that examines network traffic and alerts system administrators if there are security breaches. In this paper, an enhanced version of pigeon-inspired optimization (PIO) is proposed which enhances PIO by adding a local search algorithm named (LS-PIO). Moreover, an ensemble learning approach, which is based on multiple one-class classifiers, has been used in order to improve the performance of proposed NIDS. Four benchmark datasets were used to evaluate the LS-PIO and ensemble based NIDS which are BoT-IoT, UNSW-NB15, NLS-KDD and $KDDCUP^{99}$. The evaluation took into consideration F-score, accuracy, AUC, FPR and TPR. Results show that the suggested approach outperforms other NIDS techniques that are selected from state-of-the-art relevant research found in the literature.

1. Introduction

The global virtual environment that facilitates the sharing and exchanging of electronic resources and data over the Internet (Almobaideen & Altarawneh, 2020; Mbanaso & Dandaura, 2015) called cyberspace. Any malicious act that intents to steal information damage or resources is referred to as a cyberthreat Abu et al. (2018) and Qatawneh et al. (2020). The Internet of Things (IoT) is an extension of the Internet that allows computing devices embedded in common objects to send and receive data (Abualghanam et al., 2019). This will result in a huge amount of exchangeable data over the IoT which can be vulnerable to cyberthreats (Gopalan et al., 2021). The diversity and heterogeneity of IoT components make IoT systems security more crucial. In order to protect IoT data against different types of cyberthreats, countermeasures are needed (AbuAlghanam et al., 2021).

Fig. 1 shows the IoT Architecture from Attack Wise Perspective. Cybersecurity is all about the protection of cyberspace including hardware, software, networks, servers and peripheral devices, data and information, and all other components associated with the Internet against any risk or vulnerability. An intrusion detection system (IDS) is a sub-branch of cybersecurity that aims to protect the Internet-connected systems from both internal as well as external threats and cybercriminals. These systems identify malicious activities and isolate them from the normal traffic data (AbuAlghanam et al., 2022; Imrana et al., 2021; Sohn, 2020).

Nowadays, With the extensive use of the Internet in our daily lives and the rising amount of cyber-attacks, IDS are required to handle security challenges (Aydın et al., 2022). For that end, anomaly based detection approach can detect anomalies, and raise alerts. They are also able to detect zero-day attacks which cannot be detected using a signature-based system. On the other hand, the enormous volume of data that must be examined, as well as the difficulty in determining the borders between normal and assaults, are all problems for building intelligent network-based IDS (NIDS). In order to reduce the complexity of building intelligent NIDS we can reduce the number of features that are used to classify attacks. Feature selection is an important process to build an efficient and more accurate intelligent Intrusion Detection System (IDS) (Zhou et al., 2022).

Feature selection, as a key stage in data preprocessing, has become a popular research direction. Feature selection can also remove irrelevant features or attributes while keeping other relevant features that significantly affect data classification. As a result, feature selection may enhance the classification accuracy and speed up the model

* Corresponding author.
E-mail addresses: O.abualganam@ju.edu.jo (O. Abu Alghanam), almobaideen@inf.ju.edu.jo, wxacad@rit.edu (W. Almobaideen), M.saadeh@mdx.ac.ae (M. Saadeh), adwanoy@ammanu.edu.jo, adwanoy@ju.edu.jo (O. Adwan).

https://doi.org/10.1016/j.eswa.2022.118745
Received 24 January 2022; Received in revised form 27 August 2022; Accepted 30 August 2022
Available online 5 September 2022

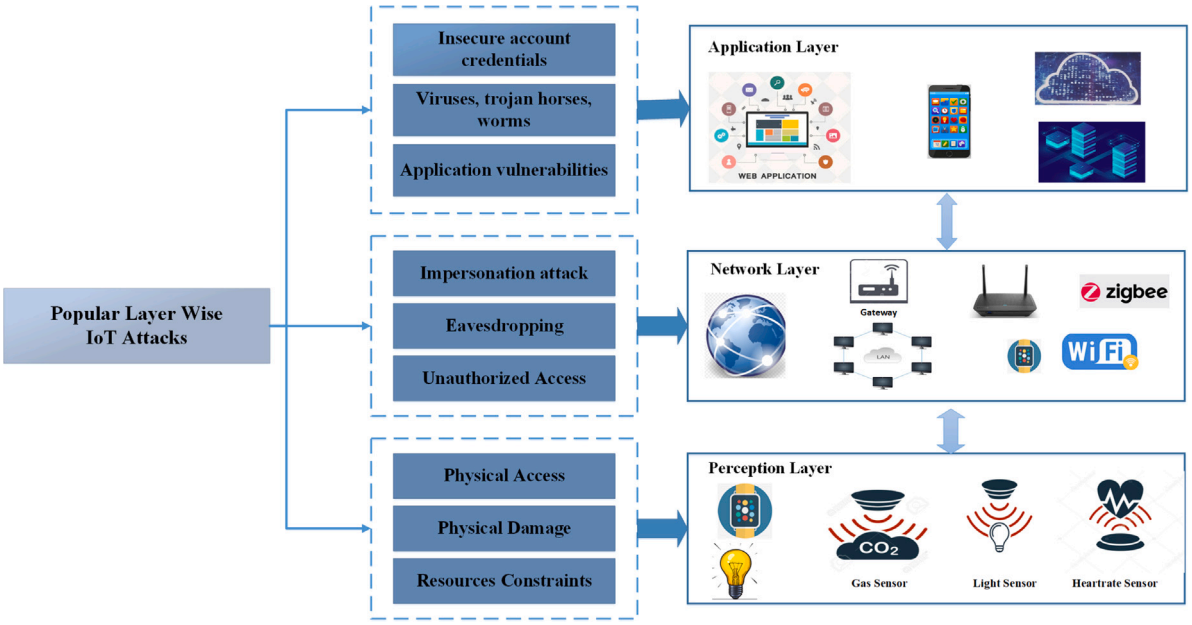

Fig. 1. IoT based Architecture from Attack Wise Perspective.

learning and data processing phase that generates the decision (Naseri & Gharehchopogh, 2022; Zhou et al., 2021).

Ensemble learning is a machine learning technique to solve a computer intelligence problem by combining numerous classifiers. Ensemble learning is widely used to improve a model's performance, such as classification, prediction, and function approximation (Alsahaf et al., 2021). It is sometimes used to lessen the risk of an unintentional poor model selection. A Bagging or Booting Aggregation can be used for Ensemble techniques (Breiman, 1996; Medina-Pérez et al., 2017).

In this paper, an intrusion detection system for anomaly detection in IoT security is proposed. The proposed IDS is based on an enhanced version of pigeon-inspired optimization (PIO) (Duan & Qiao, 2014) by integrating local search algorithm that could improve the feature selection algorithm. The main contributions of this paper are:

1. Proposing and evaluating a new one class feature selection method, named LS-PIO, that is based on the Pigeon Inspired Optimizer integrated with Tabu local search algorithm. LS-PIO helps in achieving higher accuracy and better good fitting of the results.
2. Proposing an improved lightweight ensemble based network IDS that uses LS-PIO. This anomaly detection IDS integrates one class classifier, i.e. OC-SVM, iForest, and LOF. LS-PIO is lightweight since K-Means has been used as pre-processing stage to reduce the running time of the method. This implies that this NIDS can suite IoT environment.

Topics in the rest of this paper are organized in following way. The state-of-the-art related studies on this topic are summarized in Section 2 The Mathematical model of Pigeon Inspired Optimizer is discussed in Section 3. Section 4 discusses the Proposed Model in details. Section 5 presents the Experimental results and discussion, and the conclusion is discussed in Section 6.

2. Related works

Anomaly detection techniques, known also as outlier detection, are a technique for identifying data points in a collection that differ from the norm (Ferrag et al., 2020). Since anomaly detection is used to detect patterns in data that do not conform to normal behavior, it can be used in various range of application in cybersecurity such as cyberattacks detection (Asassfeh et al., 2020; Chandola et al., 2009). Due to the high dimensionality nature of cyberattacks datasets, selecting the optimal subset of features that are used to classify attacks is crucial to the performance of the intrusion detection system. in the next subsection, some feature selection methods will be discussed.

2.1. Feature selection methods

Feature selection methods are classified into Filter, wrapper, and embedded methods (Alzaqebah et al., 2021; Wu et al., 2019). Both wrapper and embedded methods require a learner algorithm for feature selection whereas filter methods are independent of any learner algorithm (Alzaqebah et al., 2021; Ma et al., 2019). In Wan et al. (2016), a feature selection method based on a modified binary coded ant colony algorithm combined with a genetic algorithm is proposed in which each feature is treated as a binary bit to be selected or deselected. SVM is used for classification and the method is applied over multiple datasets in different domains. The authors in Alzaqebah et al. (2021) have proposed a feature selection method based on particle swarm optimization with Late acceptance hill-climbing local search. The method is evaluated based on multiple datasets obtained from UCI data source (Bache & Lichman, 2013). Another feature selection method based on particle swarm optimization are proposed in Feng and Gong (2022) and Wu et al. (2019). A combination between the filtering method and an improved quantum-behavior particle swarm optimization algorithm is utilized in Wu et al. (2019) to reduce the number of features while maximizing the accuracy. The method in Wu et al. (2019) was evaluated on 4 different gene datasets and 36 UCI datasets (Bache & Lichman, 2013) in different domains with SVM, naive Bayes, and K-nearest neighbor are used as classifiers. Another feature selection method with local search algorithm is proposed in Ma et al. (2019). This method is based on the forest optimization algorithm with tree classifiers which are support vector machine, decision tree and k-nearest neighbor. The method is evaluated using 10 different datasets obtained from the UCI data source (Bache & Lichman, 2013) and results show that decision tree achieves the best performance among other classifiers.

For intrusion detection systems, a review on feature selection methods was conducted in Bouzoubaa et al. (2021). The review evaluates different feature selection methods based on relevant DOS-DDOS datasets commonly used by several IDS research projects which are the

KDD'99 dataset (Lippmann et al., 1998), the NSL-KDD dataset (Revathi & Malathi, 2013), the UNSW-NB15 dataset (Moustafa & Slay, 2015), and the ACCS dataset (Bouzoubaa et al., 2021). In Kang and Kim (2016) a feature selection method based on a meta-heuristic local search algorithm proposed for intrusion detection systems to detect denial service attacks is discussed. The method is evaluated using the NSL-KDD dataset with a multi-layer perceptron as a classifier. The Best accuracy of 0.9937 was achieved for a subset of 25 features out of 41. Another feature selection method for IDS is proposed in Vijayanand and Devaraj (2020) which is based on an improved Whale Optimization Algorithm combined with a Genetic Algorithm to optimize the selected set of features. The method was evaluated using the CICIDS2017 and ADFA-LD standard datasets with SVM as a classifier. Best results were achieved for a subset of 35 features out of 77 original features in the CICIDS2017 dataset with about 0.96 accuracy and 25 features for the ADFA-LD out of 44 original features with about 0.94 accuracy. In Eesa et al. (2015), the authors have proposed a feature selection method based on the cuttlefish algorithm search algorithm with a decision tree classifier for intrusion detection systems. Best accuracy of 0.928 was achieved for 10 selected features out of 41 when using the benchmark $KDDCUP^{99}$intrusion dataset.

2.2. One-class classifier based feature selection methods for anomaly detection

On-class classification is the process of data classification when the dataset includes only a single label, so all data samples belong to one class (Aguilar et al., 2021; Camiña et al., 2019; Medina-Pérez et al., 2017). One-class support vector machine (OC-SVM), Isolation Forest (iForest), and Local Outlier Factor (LOF) are well-known outliers detection algorithms that are highly recommended for anomaly detection (Cheng et al., 2019; Rajasegarar et al., 2008). A comparison between the three methods is discussed in Pérez et al. (2021). Another comparison between the three methods is presented in Pérez et al. (2019) in which the authors enhance the accuracy of these methods by combining them with feature learning methods such as Principal Component Analysis (PCA) and autoencoders. According to the results, OC-SVM shows the best accuracy improvement when combined with autoencoders. Table 1 lists recent studies that use OC-SVM, IF, or LOF for anomaly detection.

2.2.1. One-class support vector machine based IDS

One-class support vector machine (OC-SVM) is a machine learning algorithm that is highly recommended for anomaly detection and there are several proposals have been used in NIDS (Rajasegarar et al., 2008). In Kittidachanan et al. (2020) a grid search one-class support vector machine algorithm (GS-OC-SVM) has been proposed and performed over German credit card and European cardholder credit card transactions datasets. The results turn out that one-class SVM produces a high true positive rate while the FScore for European cardholder dataset was 90.32 and 91.80 for GS-OC-SVM and Isolation Forest, respectively. Another study that is based on OC-SVM is found in Xiong and Zuo (2020) in which OC-SVM is used in Recognizing multivariate geochemical anomalies in a hybrid model that combines unsupervised deep belief networks (DBNs) and one-class support vector machine (OC-SVM). In Maglaras and Jiang (2014) the K-OC-SVM is presented which combines OC-SVM with RBF kernel and recursive k-means clustering. the solution is proposed for the supervisory control and data acquisition (SCADA) systems (Maglaras et al., 2016). Their model was carried out using a trace file extracted from a typical wireless network, which contains 10.000 lines, each representing a packet transferred across the network. In Alazzam et al. (2021), the authors proposed an IDS that integrates two main subsystems which collaborate, coordinate their functionality and are trained using OC-SVM. Another hybrid approach based on OC-SVM is found in Khraisat et al. (2020) through the integration of the C5 decision tree classifier and the OC-SVM. the results showed that the hybrid approach outperforms the other well-known approaches such as Naïve Bayes, random forest, multi-layer perceptron, SVM, and K nearest neighbor.

2.2.2. iForest based IDS

Another algorithm that is used for anomaly detection is Isolation Forest or iForest which has been proposed by Liu et al. (2008). It has been noticed that the use of isolation is shown to be highly effective in detecting anomalies with extremely high efficiency. In terms of AUC and execution time, the empirical study demonstrates that iForest outperforms a near-linear time complexity distance-based technique, ORCA, LOF, and RF, especially in big data sets. Furthermore, with a small ensemble size, iForest converges quickly, allowing it to discover abnormalities with great efficiency. Moreover, with an additional attribute selector, iForest can achieve good detection performance fast for high-dimensional situations with a large number of irrelevant attributes. several studies have used isolation Forest in Anomaly Detection. In Carletti et al. (2020) a Depth-based Isolation Forest Feature Importance (DIFFI) method is proposed. This method is a global interpretability method that generates Global Feature Importance (GFIs). The comparison has been done between (DIFFI), Laplacian Score (lapl) and SPEC (spec) methods. The results turn out that F-score was the highest for the DIFFI method.

2.2.3. LOF based IDS

Local outlier factor algorithms are used for outlier and anomaly detection in big data streams (Alghushairy et al., 2021). In Paulauskas and Bagdonas (2015) an anomaly detection approach based on LOF is proposed. The method is applied on fifteen different groups of features in order to detect anomalous network flows. A combination between LOF and iForest is proposed in Cheng et al. (2019). The iForest is used to find the anomaly score of each data point in the construction. Then a pruning strategy is applied to reduce the complexity and generates an outlier candidate set, and finally, LOF is applied to enhance the result and get more accurate outliers. In this paper, NIDS is proposed based on the ensemble learning approach which combines the OC-SVM, iForest, and LOF algorithms. In order to enhance the performance of the classification. On the other hand, the proposed NIDS uses the best selected features that are determined by the proposed feature selection algorithm enhanced version of pigeon-inspired optimization (Duan & Qiao, 2014) by adding local search algorithm.

2.3. Problem understanding

Up to our knowledge and based on an extensive literature review search, we noticed that most of the research has been conducted with multi-class classifiers for feature selection algorithms with a focus on reducing the TPR and FPR. Other researchers have focused on reducing the number of used features regardless of the performance. In this research, we have focused on reducing the number of features, and the performance of classification while using one class classifier. Using one class classifier means that our proposed NIDS allows for a highly accurate classification of traffic for an IoT based network that has been recently set and only normal data is available to train the IDS and get the baseline model.

3. Mathematical model of pigeon inspired optimizer (PIO)

The pigeon-inspired optimization (PIO) algorithm is a new swarm intelligence technology that is inspired by pigeon homing behavior (Chen et al., 2019). Pigeon is a kind of bird taht was used to transfer letters between people separated by long distance. Pigeons have the ability to navigate geographically to their homes. They use various parameters such as the sun, the Earth's magnetic field, and landmarks to navigate to reach the destination (Sun & Duan, 2014).

Guilford and other in Guilford et al. (2004) developed PIO algorithm that meets the pigeons behavior which is designed based on the two main operators that are used by Pigeons. The first behavior is how the pigeons determine their itinerary using the sun compass and the magnetic particles to adjust their direction. The second behavior is

Table 1
Comparison between different anomaly detection techniques proposed in literature.

Reference	Year	Algorithm category	Best results %	Dataset
Kittidachanan et al. (2020)	2020	OC-SVM	62.59(AUC)	German credit card
Kittidachanan et al. (2020)	2020	OC-SVM	92.28(AUC)	European cardholder
Kittidachanan et al. (2020)	2020	GS-OCSVM	64.71(AUC)	German credit card
Kittidachanan et al. (2020)	2020	GS-OCSVM	92.28(AUC)	European cardholder
Kittidachanan et al. (2020)	2020	iForest	62.96(AUC)	German credit card
Kittidachanan et al. (2020)	2020	iForest	93.01(AUC)	European cardholder
Xiong and Zuo (2020)	2020	OC-SVM	86(AUC)	geochemica dataset
Xiong and Zuo (2020)	2020	PCA-OCSVM	90(AUC)	geochemica dataset
Xiong and Zuo (2020)	2020	DBN-OCSVM	92.5(AUC)	geochemica dataset
Alazzam et al. (2020)	2021	OC-SVM	96.9(Acc)	KDDCUP 99
Alazzam et al. (2020)	2021	OC-SVM	83 (Acc)	NLS-KDD
Alazzam et al. (2020)	2021	OC-SVM	52.9 (Acc)	UNSW-NB15
Khraisat et al. (2020)	2020	C5	81.53 (Acc)	NSL-KDD
Khraisat et al. (2020)	2020	C5	97.3(Acc)	ADFA
Khraisat et al. (2020)	2020	OC-SVM	72.17(Acc)	NSL-KDD
Khraisat et al. (2020)	2020	OC-SVM	76.40(Acc)	ADFA
Khraisat et al. (2020)	2020	C5-OCSVM	83.24(Acc)	NSL-KDD
Khraisat et al. (2020)	2020	C5-OCSVM	97.4(Acc)	ADFA
Tian et al. (2018)	2018	OC-SVM	95.61(Acc)	NSL-KDD
Tian et al. (2018)	2018	OC-SVM	95.44(Acc)	UNSW-NB15
Tian et al. (2018)	2018	Ramp-OCSVM	98.59(Acc)	NSL-KDD
Tian et al. (2018)	2018	Ramp-OCSVM	97.24(Acc)	UNSW-NB15
Paulauskas and Bagdonas (2015)	2015	LOF	81(F1)	VilniusGediminas Technical University Faculty of Electronics Network flow
Cheng et al. (2019)	2019	iForest	99.26(Acc)	EMGPA
Cheng et al. (2019)	2019	iForest	97.69(Acc)	KEGG
Cheng et al. (2019)	2019	iForest	99.09(Acc)	EEGES
Cheng et al. (2019)	2019	LOF	97.44(Acc)	EMGPA
Cheng et al. (2019)	2019	LOF	92.40(Acc)	KEGG
Cheng et al. (2019)	2019	LOF	99.85(Acc)	EEGES
Cheng et al. (2019)	2019	iForest-LOF	99.70(Acc)	EMGPA
Cheng et al. (2019)	2019	iForest-LOF	99.74(Acc)	KEGG
Cheng et al. (2019)	2019	iForest-LOF	99.99(Acc)	EEGES

landmark-based with allows a pigeon to be guided by landmarks if they are familiar with the location or, otherwise, follow the leader who is familiar with the landscape of their homing.

Two operators are constructed to idealize some of the homing qualities of pigeons which are as follows:

1. **Map and compass operator:** in this phase each pigeon i has a position X_i and velocity V_i which are represented by a D-dimension search space. Moreover, using $(t+1)$th the position and the velocity will be updated in the next iteration. The update is depending on the value of the current iteration tth as in Eqs. (1) and (2). Moreover, rand represents a uniform random number within the range [0,1], R represents the map and compass factor and X_g represents the current global best position that can be obtained by comparing all the positions among all the pigeons (Duan & Qiao, 2014).
As a feature selection algorithm the position X_i represents the solution that holds a group of selected features. Moreover, the velocity V_i represents the amount of change toward the best pigeon.

$$V_i(t) = V_i(t-1).e^{-Rt} + rand.(X_g - X_i(t-1)) \quad (1)$$

$$X_i(t) = X_i(t-1) + V_i(t) \quad (2)$$

2. **Landmark operator:** When the pigeons are getting close to their goal, they will rely on nearby landmarks. All pigeons are ranked based on the fitness value then at each iteration, the number of pigeons will be divided over two to be the half as in Eq. (3) where N_p represents the number of pigeons in t which is the current iteration. Moreover, all other pigeons use Eqs. (4) and (5) to update their position to best position they can find, X_i represents the current position of all pigeons and X_c represents the position of the centered pigeon (best destination).

$$N_p(t+1) = \frac{N_p(t)}{2} \quad (3)$$

$$X_c(t+1) = \frac{\sum X_i(t+1).Fitness(X_i(t+1))}{N_p \sum Fitness(X_i(t+1))} \quad (4)$$

$$X_i(t+1) = X_i(t) + rand.(X_c(t+1) - X_i(t)) \quad (5)$$

The fitness function represents the quality of the pigeon individual solution where the maximum or the minimum value can be the target. The number of selected features, False Positive Rate (FPR) and True Positive Rate (TPR) is used to evaluate the fitness function in our model.

4. The proposed model

In this paper, a modified version of PIO feature selection algorithm named Local Search PIO (LS-PIO) has been proposed. LS-PIO is based on two local search algorithms, i.e. Hill Climbing and Tabu search. Moreover, an ensemble method has been used for the proposed general network-IDS that is based on three one-class classifiers as shown in Fig. 2.

4.1. Fitness function

The main objective of the fitness function that is used in the LS-PIO, see line 18 36, is to calculate the fitness value for each solution that is generated in each iteration and for the solutions generated during the map and compass phase. In our case, the less the value of the fitness function the better the solution. Eq. (6) represents the mathematical equation used in the fitness function. N represents the total number of features while F represents the whole number of selected features. Moreover, $\alpha = 0.06$, β and $\delta = 0.48$ should sum to 1 and represent weights for F, FPR, and TPR respectively.

$$FF = min[\alpha * \frac{F}{N} + \beta * FPR + \delta * \frac{1}{TPR}] \quad (6)$$

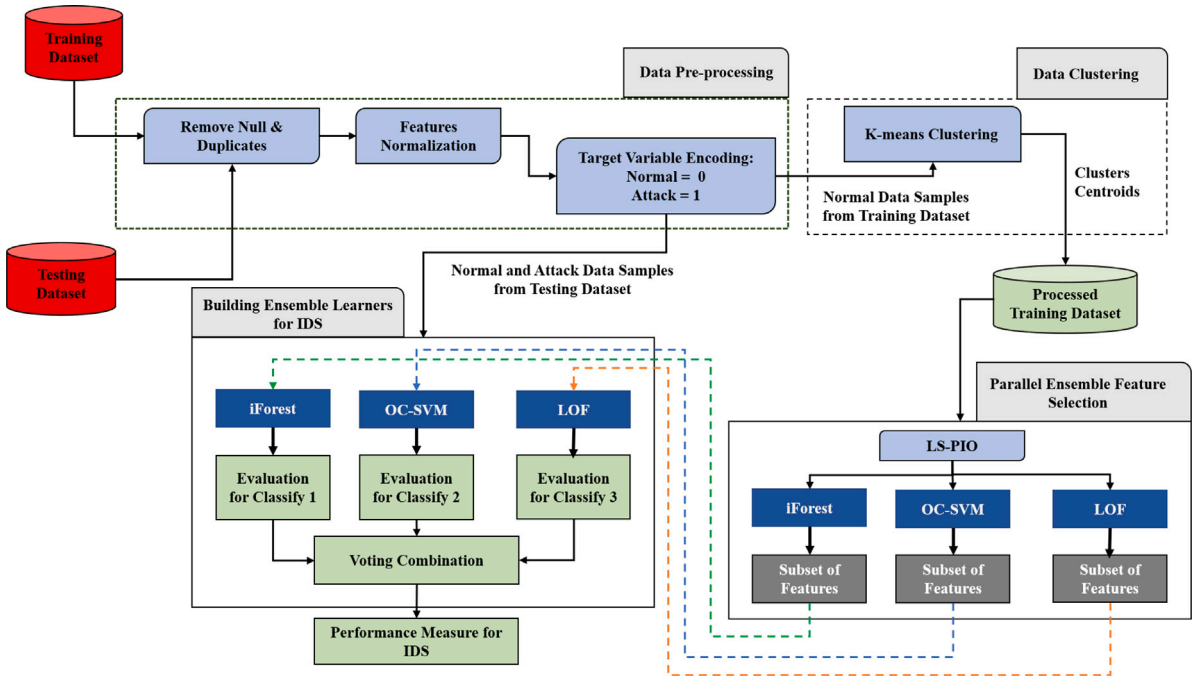

Fig. 2. Architecture of LS-PIO based NIDS.

The proposed feature selection algorithm LS-PIO uses three one-class classifiers, i.e. iForest, OC-SVM, and LOF. Each classifies produces a set of selected features, for a certain dataset, that will be used by the corresponding IDs during the process of building ensemble learners.

4.2. Data pre-processing

Dataset pre-processing is a process that includes data normalization, reduction, cleaning, and transformation. These steps are important and can affect the classifier performance (da Costa et al., 2021)). In this paper, we have applied the following steps for data pre-processing as shown in Fig. 2. Firstly, remove null values and remove duplicated records to prevent any bias to these most common records. For the $KDDCUP^{99}$ training set, the original number of records is 494,109 after removing all duplicates, the records become 145,584. On the other hand, NSL-KDD and UNSW-NB15 training sets are not having any duplicates records. Next, data normalization is applied through scaling the data values into a proportional range of each feature. This step is crucial to avoid dataset's bias toward features that inherently has larger values (Patro & Sahu, 2015). According to Eq. (7), all datasets values have been normalized into the range [0,1].

The next step is to convert the symbolic data into numeric values. In all datasets the attack types have been transformed into a label that is set to "1", while the normal instance's label has been set to "0". Then, for the training datasets, only normal data samples are extracted from the datasets. This is due to that fact that the proposed IDs is supposed to be use in the early stages of a network lifetime, where the normal traffic is the only type that should be learnt. Then, in the second stage (testing), the proposed IDs is supposed to differentiate between normal and Attack traffic. Finally, for training stage, the extracted normal samples are clustered in order to reduce the number of records and accelerate the training speed.

$$X_{normalized} = \left(\frac{X - X_{min}}{X_{max} - X_{min}}\right) \tag{7}$$

4.3. Data clustering

The feature selection phase relies on the LS-PIO algorithm which very demanding in terms of computation power. Additionally, OC-SVM and LOF classifiers are invoked during the feature selection phase and this adds to the computation complexity as mentioned in section . To address this issue, k-means clustering can be utilized to generate a representative smaller in size dataset from each of the considered datasets.

K-means clustering is based on choosing a random centroid for each cluster from a set of k clusters. Based on their distance or similarity to each centroid, the k-means algorithm assigns data points to the nearest cluster. After all data points have been assigned to the group that is closest to each point, a centroid point is calculated for each cluster. The centroid represents the average of all the data points in a certain cluster. The procedure for allocating data points to the new cluster's centroid is then repeated, and the centroid is recalculated until the values of the centroid stabilize.

In this paper, K-Means clustering has been performed only for the training datasets. The main goal of clustering these datasets is to reduce the number of records in each dataset by replacing a set of similar records with one record, i.e. the cluster head or the centroid. The resulted records represent the original dataset but enhance the speed of classifier processing. Fig. 3 presents the runtime for UNSW-NB15 dataset as the optimal number of clusters is increased from 50, 100, 250, 500, and up to 3000. One can notice that we can achieve reasonable run time for all number of clusters until we exceed 500 after which the run time jumps sharply. Similar results have been achieved for $KDDCUP^{99}$, NSL-KDD. For BoT-IoT we did the same experiment and the result was that the optimal number of clusters is 1000 which has been used in order to run the feature selection algorithms.

4.4. Local search based pigeon inspired optimizer LS-PIO model

The importance of a feature selection process which has a few desired purposes, the most important of which is to reduce the dimensionality of the dataset by removing duplicate and irrelevant characteristics. Moreover, selecting the most important features that have a strong correlation with the target class. Resolving all these issue boost performances and cuts down on processing time. In this paper, a new modified version feature selection called LS-PIO is proposed as shown in Algorithm 36.

Algorithm 1 Modified Pigeons inspired optimizer LS-PIO

Input: Population size N_p, Space Dimension D, Map and compass factor R,Number of local search iterations, Number of pigeons iterations nc_1, nc_2 where $nc_1 > nc_2$
Output: Global Solution X_g

Initialize X_i for each Pigeon randomly.
Evaluate Pigeons $(X_1, X_2, \ldots, X_{N_p})$ by their fitness values.
Xg = best pigeon (Minimum fitness)
while (n_c >= 1) **do**
    Update velocity and path for each pigeon by Eqs. (1) and (2).
    Evaluate Pigeons $(X_1, X_2, \ldots, X_{N_p})$ by their fitness values.
    Update X_g
    local_search(X_g, hill_climbing)
    (Update X_g)
end while
while (N_p >= 1) **do**
    Sort pigeons by their fitness values.
    $Np = N_p/2$
    Calculate desired destination by Eq. (4)
    Update pigeon position by Eq. (5).
    Update X_g
end while
Xg = best pigeon (Minimum fitness)
Xg = Local Search
Xg = best pigeon from Local Search (Minimum fitness)
while (n_c >= 1) **do**
    Update velocity and path for each pigeon by map and compass operator.
    for each Pigeon (X_i *in* N_p) **do**
        Train Isolation forest on (X_i)
        Evaluate Pigeons (X_i) by their fitness values.
    end for
    Return best Isolation forest ν, γ and pigeon.
    Update X_g
end while
while (N_p >= 1) **do**
    Sort pigeons by their fitness values.
    $Np = N_p/2$
    Calculate desired destination
    Update pigeons position's toward the desired destination.
    Update X_g
end while

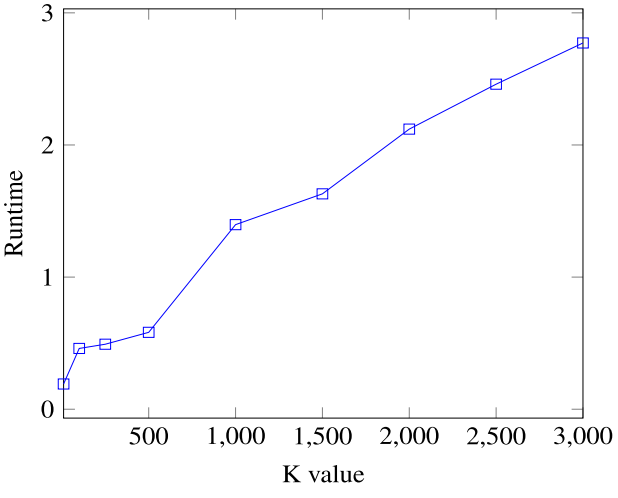

Fig. 3. Optimal number of clusters for UNSW-NB15 dataset.

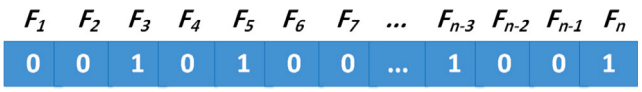

Fig. 4. LS-PIO binary bit representation for the input Dataset.

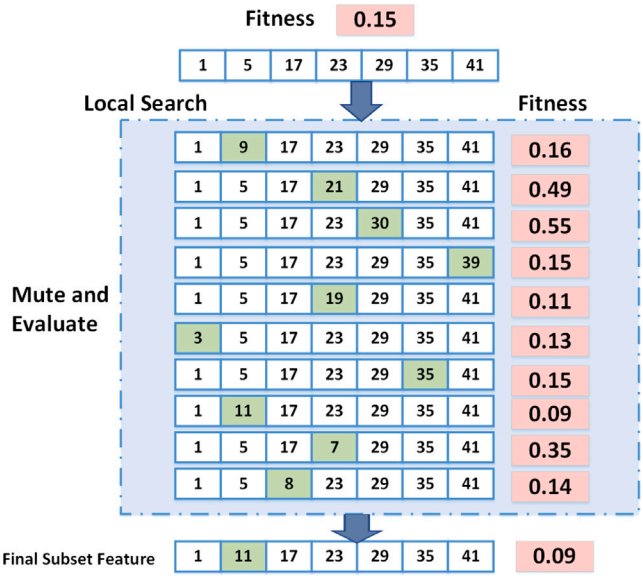

Fig. 5. Feature selection update using local search for 10 iteration.

In LS-PIO, a random solution will be generated which is based on Np, each solution is a vector that holds the different number of features. The solution's value is set to a binary value of zero or one at random. A zero number indicates that the relevant characteristic is not present in the current solution, whereas one value shows that the relevant feature is present. A randomly generated solution for a dataset with a variable number of characteristics is shown in Fig. 4

The best solution will be determined which is called a global solution on the other hand, the rest of the pigeons (solutions) will modify to word the best one or reach better then the best pigeon. Eq. (8) demonstrates how to determine pigeon velocity. The velocity of each pigeon that represent a solution is updated by determining the degree of similarity between the solutions that represent each pigeon and the global solution, resulting in a unique velocity value for each pigeon or solution.

To compute velocity, the cosine similarity formula in Eq. (8) is utilized to calculate the similarity ratio between the global pigeon Xg and local pigeon Xp. Based on the result the velocity value will update the pigeon's position. According to Eq. (9) the pigeon's position will be adjusted based on its probability of being comparable to the global solution.

After each iteration the best global solution will be entered in the local search and in this paper two local search algorithms have been used which are hill climbing and tabu search. To find another solution based on mutating the current solution to find another better one. Fig. 5 shows how the best solution will update to reach another one if it has been found otherwise the old solution will be returned.

4.4.1. Modified map and compass operator

PIO algorithm was created to deal with continuous data based on Eq. (1) in Alazzam et al. (2020) an adaptive version of PIO to deal with the basic process for map and compass operator. The primary purpose of this stage is to update the pigeon position. This is done based on the velocity and position of the swarm's best pigeon. This

was accomplished by subtracting the local pigeon's position Xi from the global or best pigeon's position.

Eq. (9) will update the position of the pigeon based on the velocity value V_p from Eq. (8). The pigeon's position will be updated based on the probable similarity with the global solution, according to Eq. (9). In addition, Hill-Climbing and Tabu Search are utilized to improve the algorithm in order to obtain a better solution.

$$V_p = Cosine\ Similarity(X_g, X_p) = \frac{X_g.X_p}{\|X_g\|.\|X_p\|} = \frac{\sum_{i=0}^{n-1} X_{p,i} X_{g,i}}{\sqrt{\sum_{i=0}^{n-1} X_{p,i}{}^2}\sqrt{\sum_{i=0}^{n-1} X_{g,i}{}^2}} \tag{8}$$

$$X(t)_{(i,p)}[i] = \begin{cases} X(t-1)_p[i], & if(S(V_i(t)) > r) \\ X(t-1)_g[i], & \text{otherwise} \end{cases} \tag{9}$$

Fig. 4 shows an example of how the global solution will update using the local search algorithm based on the fitness value and the number of iterations.

4.4.2. Modified landmark operator

The landmark operator consists of two main operations the first one is determining the desired pigeon destination, which is identical or close from the base one. The fitness value is used to rank all pigeons. The number of pigeons will be updated by Eq. (3) in each generation, and only half of them are used to find the intended position of the centered pigeon. The rest of the pigeons, on the other hand, modify their intended destination by following the desired destination position. Eq. (3) is used to calculate the position of the target destination.

In the second phase of the landmark operation after all pigeons modify their positions to reach the intended destination which will be different from the binary vector. Consequently, in Eq. (8) all pigeons will be updated their positions initially by determining their velocity and according to Eq. (9) then all pigeons have updated their positions.

4.5. Ensemble model

The output of the feature selection phase is fed into the next phase which is the ensemble IDS. Specifically, each feature selection method's output that is based on the LS-PIO in fed into the corresponding method in the ensemble IDS. For example, features selected by LOF during the feature selection phase are fed into the LOF method which is part of the ensemble IDS. Accordingly, the ensemble IDS combines three one-class classifiers which are one class support vector machine (OC-SVM), Isolation Forest (IF), Local Outlier Factor (LOF) into one ensemble method. The final classification result for the testing data point is based on the weighted vote of the output produced by three classifiers.

4.6. Complexity analysis

The run time complexity for the NIDS is based on the time complexity of the two subsystems which are featured selected and intrusion detection subsystems. The feature selection includes clustering the data, feature selection using LS-PIO and model training process and validation. The IDS subsystem is based on ensemble learning that consists of three one-class classifiers run in parallel, each classifier consists of the training part, and the testing part.

- Data Clustering complexity
  Data clustering has been applied over the training dataset to prepare for using it in the two subsystems. Eq. (10) illustrated the time complexity for the data clustering that is based on the following parameters. K is the number of clusters that are needed. F is the number of features. T represents the time period needed to find the distance between two data points. Finally, i is the number of iterations needed for convergence of the k-means algorithm.

$$O(C) = O(K) * O(F) * O(T) * O(i) \tag{10}$$

- Feature Selection Complexity
  The initialization phase of the population, map and compass operator, local search, and landmark operator that are utilized to generate a new solution are the key stages of the algorithms that define the run time complexity of LS-PIO. Eq. (11) presents the complexity for LS-PIO.

$$\begin{aligned} O(LS-PIO) = {} & O(Initialization\ phase) \\ & + O(runtime\ of\ map\ and\ compass) \\ & + O(runtime\ of\ Local\ Search) \\ & + O(runtime\ of\ landmark\ operator) \\ & + O(fitness\ function) \end{aligned} \tag{11}$$

  The complexity of the initialization phase is based on the number of initialized pigeons which equals to O(N_p) while, the complexity of the map and compass operator is based on updating each pigeon toward best pigeon. The complexity for Hill-Climbing and Tabu search are O(N_i) where i represents the number of iterations.
  The landmark operator's complexity is high which equals O($N_p \log_2 N_p$) since in each iteration, the number of pigeons will be decreased to half.
  The complexity of the fitness function relies on the wrapper function complexity, parameter tuning, and predicting. In the proposed modal three classifiers were used each time which is iForest, LOF and OC-SVM as we will discuss the complexity for each in detail in the section ensemble learning complexity.
- Ensemble Learning complexity
  The Ensemble technique has been used in IDS the complexity analysis has been divided into two parts which are training and testing. The time complexity of IDS is equal to the highest run time between the three classifiers which is the runtime for OC-SVM. Eq. (12) illustrated the complexity runtime for ensemble approach.

$$\begin{aligned} O(EM) = {} & O(OC-SVM) + O(iForest) \\ & + O(LOF) + O(Voting) \end{aligned} \tag{12}$$

  In OC-SVM the training time has been repeated 10 times, which equals to the number of ν in νs array. The fitness function time complexity for each pigeon becomes O($4 * (k^2 N_p + k^3) + N_p$). The overall run time complexity is presented in Eq. (13).

$$\begin{aligned} O(LS-PIO) = {} & O(N_p) + O(N_p * t_1) \\ & + O((N_p \log_2 N_p) * t_2) + O(k^2 N_p + k^3 + N_p) \\ = {} & O(k^2 N_p + k^3 + N_p) \end{aligned} \tag{13}$$

$$\begin{aligned} O(OC-SVM) = {} & O(N_p) + O(N_p * t_1) \\ & + O((N_p \log_2 N_p) * t_2) + O(k^2 N_p + k^3 + N_p) \end{aligned} \tag{14}$$

  In iForest the time complexity is based on the F — input dataset, T — number of trees, N — subsampling size as illustrated in Eq. (15).

$$\begin{aligned} O(iForest) = {} & O(F) + O(T) \\ & + O(N) \end{aligned} \tag{15}$$

  The computational complexity of local outlier factor as shown in Eq. (16) is based on five definitions which are k-distance of a data point p, k-Nearest Neighbors of p, a reachability distance of p with respect to, local reachability density of p and LOF with p as detailed in Alghushairy et al. (2021).
  The complexity runtime for the Voting is O(A) while runtime complexity for the ensemble learning is based on the highest runtime which is for OC-SVM.

$$O(LOF) = O(N^2) \tag{16}$$

Table 2
No. of records of training and testing distribution for datasets.

No	Category	UNSW-NB15		BoT-IoT		KDDCUP99		NSL-KDD	
		Train set	Test set	Train set	Test set	Train set	Test set	Train set	Test set
1	Normal	37,000	56,000	370	1079	97,277	31,052	67,343	9710
2	DoS	4089	12,264	1,320,148	330,112	391,458	99,904	45,927	7457
3	DDoS	–	–	1,541,315	385,309	–	–	–	–
4	Probe	–	–	–	–	3015	4107	11,656	2421
5	U2L	–	–	–	–	52	86	52	200
6	Generic	18,871	40,000	–	–	–	–	–	–
7	Analysis	677	2000	–	–	–	–	–	–
8	Fuzzers	6062	18,184	–	–	–	–	–	–
9	Worms	44	130	–	–	–	–	–	–
10	Exploits	11,132	33,393	–	–	–	–	–	–
11	Backdoor	583	1746	–	–	–	–	–	–
12	Shellcode	378	1133	–	–	–	–	–	–
13	Reconnaissance	3496	10,491	72,919	18,163	–	–	–	–
14	Theft	–	–	1269	318	–	–	–	–
15	R2L	–	–	–	–	1126	4300	995	2754
	Total	82,332	175,341	2,934,817	733,705	494,109	139,246	125,972	22,542

Table 3
$KDDCUP^{99}$ and NSL-KDD features in terms of category and data type.

Category	No.	Name	Data type	Category	No.	Name	Data type
Basic	1	duration	Continuous	Content	22	is_guest_login	Symbolic
	2	wrong_fragment	Symbolic		23	srv_diff_host_rate	Continuous
	3	dst_bytes	Symbolic		24	diff_srv_rate	Continuous
	4	urgent	Continuous		25	serror_rate	Continuous
	5	src_bytes	Continuous		26	srv_serror_rate	Continuous
	6	service	Continuous		27	count	Continuous
	7	Land	Symbolic		28	srv_rerror_rate	Continuous
	8	protocol_type	Symbolic		29	same_srv_rate	Continuous
	9	Flag	Symbolic		30	srv_count	Continuous
Content	10	is_host_login	Symbolic		31	rerror_rate	Continuous
	11	num_failed_logins	Continuous	Traffic	32	dst_host_diff_srv_rate	Continuous
	12	num_root	Continuous		33	dst_host_srv_rerror_rate	Continuous
	13	num_compromised	Continuous		34	dst_host_same_srv_rate	Continuous
	14	root_shell	Continuous		35	dst_host_count	Continuous
	15	num_outbound_cmds	Continuous		36	dst_host_srv_serror_rate	Continuous
	16	logged_in	Symbolic		37	dst_host_srv_diff_host_rate	Continuous
	17	num_file_creations	Continuous		38	dst_host_serror_rate	Continuous
	18	num_shells	Continuous		39	Class	Symbolic
	19	num_access_files	Continuous		40	dst_host_same_src_port_rate	Continuous
	20	su_attempted	Continuous		41	dst_host_srv_count	Continuous
	21	Hot	Continuous		42	dst_host_rerror_rate	Continuous

5. Experimental results and discussion

In this section, we describe the datasets that use running the experiments designed to evaluate LS-PIO algorithm, presents the setup parameters of the experiments, discuss the performance metrics that are used to compare LS-PIO and its rivals and delve into the details of the comparison through a systemic discussion.

5.1. Dataset description

The LS-PIO has been evaluated using some of benchmark datasets that are used by other state-of-the-art proposed techniques such as $KDDCUPP^{99}$ (Lippmann et al., 1998), NSL-KDD (Revathi & Malathi, 2013), UNSW-NB15 (Moustafa & Slay, 2015) and BoT-IoT (Koroniotis et al., 2019).

Table 2 shows the specifications of various records of Training and Testing Distribution for each of the Datasets used in this evaluation (Moustafa & Slay, 2016).

5.1.1. KDDCUP99

In order to support research that focuses on intrusion detection, DARPA dataset has been created in 1998 and then upgraded in 1999 by publishing a newer version which is known as $KDDCUP^{99}$. The main goal of $KDDCUPP^{99}$ to allow researchers to experience with techniques that can detect and classify events as either good or bad. The four main types of attacks, i.e Denial of Service Attacks (DoS), Probing attacks (Probe), User to Root Attacks (U2R), and Remote to Local attacks (R2L) that are included in $KDDCUP^{99}$ are presented in Table 2. The features included in $KDDCUPP^{99}$ dataset with their classification under one of the main three categories, i.e. basic, content, and traffic, and their datatype, i.e. continuous and symbolic are presented in Table 3.

5.1.2. NSL-KDD

NSL-KDD dataset has been created as an improved version of $KDDCUPP^{99}$. It solves certain problems of the $KDDCUP^{99}$ such as those mentioned by Tavallaee et al. (2009). It includes the proper number of records in its training and testing tests. It maintains the original characteristics of the base dataset, namely $KDDCUP^{99}$. Moreover, it is used as a benchmark adopted by many researchers to evaluate their suggested IDSs (Revathi & Malathi, 2013). The features included in NSL-KDD dataset with their classification under one of the main three categories, i.e. basic, content, and traffic, and their datatype, i.e. continuous and symbolic are presented in Table 3.

5.1.3. UNSW-NB15

Mustafa and Slay designed UNSW-NB15 (Moustafa & Slay, 2015) using the IXIA Perfect Storm tool, which monitors regular network

Table 4
The features for UNSW-NB15 in terms of data types and categories.

Category	No.	Name	Data type	Category	No.	Name	Data type
Flow	1	dstip	Nominal	Content	25	res_bdy_len	Integer
	2	sport	Integer		26	trans_depth	Integer
	3	proto	Nominal	Time	27	synack	Float
	4	dsport	Integer		28	Djit	Float
	5	srcip	Nominal		29	ackdat	Float
Basic	6	service	Nominal		30	Ltime	Timestamp
	7	dur	Float		31	Sintpkt	Float
	8	dttl	Integer		32	Dintpkt	Float
	9	dloss	Integer		33	tcprtt	Float
	10	sttl	Integer		34	Sjit	Float
	11	Sload	Float		35	Stime	Timestamp
	12	sloss	Integer	General purpose	36	ct_ftp_cmd	Integer
	13	dbytes	Integer		37	is_ftp_login	Binary
	14	state	Nominal		38	ct_flw_http_mthd	Integer
	15	sbytes	Integer		39	ct_state_ttl	Integer
	16	Dload	Float		40	is_sm_ips_ports	Binary
Content	17	Spkts	Integer	Connection	41	ct_dst_sport_ltm	Integer
	18	Dpkts	Integer		42	ct_srv_dst	Integer
	19	swin	Integer		43	Class	Binary
	20	dmeansz	Integer		44	ct_src_ ltm	Integer
	21	stcpb	Integer		45	ct_src_dport_ltm	Integer
	22	dtcpb	Integer		46	attack_cat	Nominal
	23	smeansz	Integer		47	ct_dst_src_ltm	Integer
	24	dwin	Integer		48	ct_dst_ltm	Integer
					49	ct_srv_src	Integer

Table 5
Set of features for the Bot-IoT dataset.

	Feature	Description	Type
1	N_IN_Conn_P_SrcIP	Number of inbound connections per source IP.	Generated flow features
2	N_IN_Conn_P_DstIP	Number of inbound connections per destination IP.	
3	pkSeqID	Row Identifier	Network flow Extracted
4	proto	Textual representation of transaction protocols present in network flow	
5	saddr	Source IP address	
6	sport	Source port number	
7	daddr	Destination IP address	
8	dport	Destination port number	
9	mean	Average duration of aggregated records	
10	drate	Destination-to-source packets per second	
11	srate	Source-to-destination packets per second	
12	max	Maximum duration of aggregated records	
13	category	Traffic category	
14	seq	Argus sequence number	
15	stddev	Standard deviation of aggregated records	
16	min	Minimum duration of aggregated records	
17	state_number	Numerical representation of feature state	
18	subcategory	Traffic subcategory	
19	attack	Class label: 0 for Normal traffic, 1 for Attack Traffic	

traffic behavior and includes nine categories of contemporary attack methods. This dataset includes a mix of real-world and simulated network traffic assault actions. The dataset's features are created using both traditional and novel methods.

UNSW-NB15 contains 49 features that can be classified under several categories. Training and testing were split as illustrated in Table 2. Moreover, the data type and the categories of different features included in this dataset are flow, basic, content, time, connection and general as illustrated in Table 4.

In this paper, UNSW-NB15 dataset has been chosen for many reasons. The first one is because it is to date dataset and several studies have used it in order to work on enhancing the performance of IDS. On the other hand, several proposals in anomaly based IDS still have a challenge to enhance the performance, using this complex dataset, especially on one class classifiers (Alhajjar et al., 2021).

5.1.4. Bot-IoT

BOT-IoT dataset has 19 attributes and It contains IoT traffic of smart home, five devices which are smart garage door, weather station, smart fridge, smart thermostat and motion-activated lights (Koroniotis, 2020; Koroniotis & Moustafa, 2020). The smart garage door has been involved namely remotely smart garage door which open or close based on probabilistic input while a weather station generates information about temperature, humidity, and air pressure. Moreover, a smart fridge regulates the fridge temperature automatically when necessary also a smart thermostat which setup the home temperature by starting the air-conditioning system, and motion-activated lights turns the light on or off based on the motion-sensor signal.

It contains both benign traffic for IoT and other network traffic and has four types of cyberattacks; information theft, probing, Denial of Service. In this paper, only 5% of the BOT-IoT dataset has been used. Table 5 illustrates the set of features for the Bot-IoT in terms of description and type.

5.2. Setup of the experiments

In this section, all experiments have been conducted on a Laptop Windows 10 a 64-bit operating system and Intel Core i7, 16 GB RAM.

Table 6
The experimental setup.

Number of runs = 20	
OC-SVM Parameters	
Parameter	Value
γ	Scale
ν	[0.1, 0.01, 0.001, 0.074]
ν	[0.1, 0.01, 0.001, 0.074]
kernel	RBF
Fitness function	
α	0.06
β	0.48
δ	0.48
PIO parameters	
Local_search_iter_max	20
Map and Compass Factor	0.09
Number of Iterations	400
Population size (Np)	128

Moreover, Anaconda Python framework version 5.1 has been used to implement the NDIS and LS-PIO. Table 6 shows the initialized value of parameters that have been used in the proposed model. Moreover, the 10-fold cross-validation method was used to partition the training dataset into ten approximately equal parts. The cross-validation process was repeated 10 times. For each iteration, one part was used as the testing data, while the remaining nine parts served as the training data.

5.3. Performance metrics

The performance metrics that have been adopted to evaluate the various techniques compared in this paper are defined in this subsection along with the formulas that specify their calculation (Sohn, 2020) :

- **False Positive Rate** (FPR or False Alarms): As in Eq. (17) it calculates the percentage of normal class who are classified as attackers.

$$FPR = \frac{FP}{TN + FP} \tag{17}$$

- **Accuracy**: As in Eq. (18) it represents the fraction of correctly identified classes to the total number of classifications.

$$Accuracy = \frac{TP + TN}{TN + TP + FN + FP} \tag{18}$$

- **F-score** (F-measure): As in Eq. (19) it represents the accuracy of the model by taking into consideration both precision and recall values.

$$F - Score = \frac{2 * TP}{2 * TP + FP + FN} \tag{19}$$

- **AUC** : Area under the curve of the true positive detection rate (TP) versus the false positive detection rate (FP). This indicator gives an idea of the general accuracy of the classifier for all false positive detection rates.

5.4. Results

In this paper, the results have been evaluated using four datasets, i.e. $KDDCUP^{99}$, UNSW-NB15, BoT-IoT and NSL-KDD for the two subsystems. The first subsystem is the adapted feature selection algorithm which is based on local search in addition to binary versions of Pigeon Inspired Optimizer called LS-PIO. The LS-PIO was analyzed and compared to some of the recently proposed algorithms of feature selection.

The second subsystem evaluation for NIDS has been presented based on different performance metrics, as discussed in 5.3, and compared with other related network intrusion detection techniques.

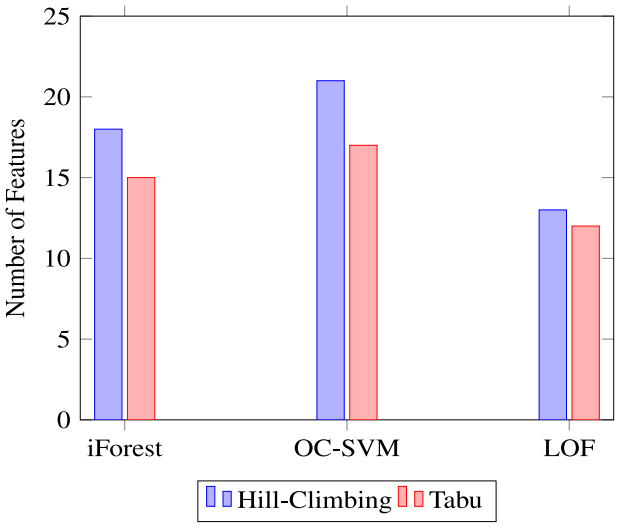

Fig. 6. The number of selected features based on LS-PIO on $KDDCUP^{99}$ dataset.

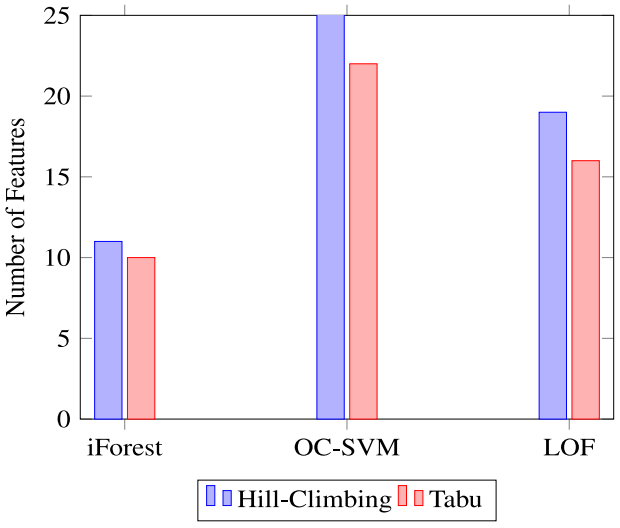

Fig. 7. The number of selected features based on LS-PIO on NSL-KDD dataset.

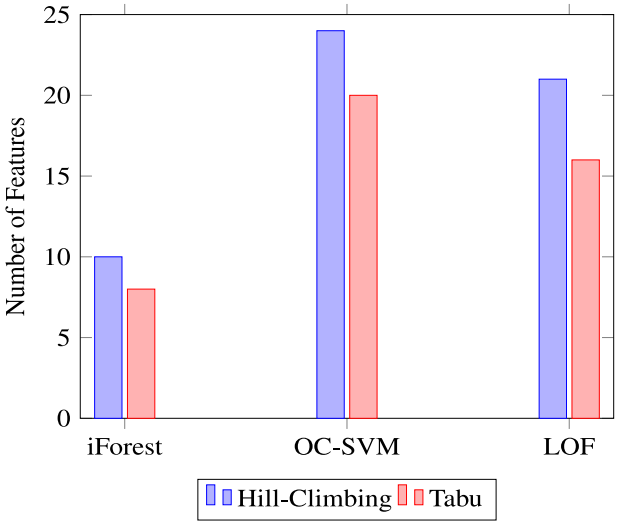

Fig. 8. The number of selected features based on LS-PIO on UNSW-NB15 dataset.

Table 7
The selected features from $KDDCUP^{99}$ by LS-PIO based on different fitness function and local search.

Classifier	Local search	Number of features	Selected features
iForest	PIO-Hill-Climbing	18	1, 2, 3, 4, 6, 7, 8, 11, 13, 17, 20, 31, 32, 33, 35, 36, 38, 39]
	PIO-Tuba Search	15	[1, 2, 3, 4, 8, 11, 20, 26, 27, 31, 32, 35, 36, 39, 40]
OC-SVM	PIO-Hill-Climbing	21	[0, 2, 3, 5, 6, 9, 11, 16, 19, 20, 21, 24, 25, 27, 28, 30, 33, 35, 37, 38, 39]
	PIO-Tuba Search	17	[3, 4, 8, 9, 10, 11, 12, 14, 15, 17, 26, 30, 33, 35, 39, 40]
LOF	PIO-Hill-Climbing	22	[2, 3, 4, 6, 8, 9, 10, 11, 12, 14, 15, 17, 20, 26, 29, 30, 33, 34, 35, 38, 39, 40]
	PIO-Tuba Search	15	[4, 6, 7, 8, 10, 12, 17, 20, 22, 24, 25, 28, 33,34,37]

Table 8
The selected features from NSLKDD by LS-PIO based on different fitness function and local search.

Approach	Local search	Number of features	Selected features
iForest	PIO-Hill-Climbing	11	[3, 5, 6, 7, 9, 11, 19, 23, 31, 32, 35]
	PIO-Tuba Search	10	[1, 3, 5, 6, 7, 9, 11, 20, 31, 32]
OC-SVM	PIO-Hill-Climbing	27	[0, 1, 3, 6, 7, 8, 12, 13, 16, 17, 18, 19, 20, 21, 24, 25, 26, 27, 29, 31, 32, 33, 34, 35, 36, 38, 40]
	PIO-Tuba Search	22	[0, 1, 6, 10, 11, 12, 13, 15, 18, 19, 24, 26, 29, 31, 32, 33, 34, 35, 37, 38, 39, 40]
LOF	PIO-Hill-Climbing	19	5, 6, 7, 11, 12, 14, 15, 16, 18, 19, 21, 26, 27, 28, 31, 32, 34, 37, 39]
	PIO-Tuba Search	16	[1, 5, 7, 10, 12, 13, 15, 17, 18, 20, 25, 31, 32, 34, 39, 40]

Table 9
The selected features from UNSW-NB15 by LS-PIO based on different fitness function and local search.

Approach	Local search	Number of features	Selected features
iForest	PIO-Hill-Climbing	10	[5, 10, 11, 17, 18, 21, 22, 30, 32, 42]
	PIO-Tuba Search	8	[10, 11, 14, 15, 17, 30, 32, 42]
OC-SVM	PIO-Hill-Climbing	24	[1, 2, 3, 6, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 27, 28, 29, 31, 34, 37, 39, 40, 42]
	PIO-Tuba Search	20	[1, 3, 4, 11, 12, 13, 15, 18, 19, 23, 24, 26, 27, 28, 30, 33, 34, 39, 40, 41]
LOF	PIO-Hill-Climbing	21	[[1, 2, 4, 5, 7, 8, 10, 11, 14, 15, 17, 24, 25, 26, 28, 29, 32, 33, 34, 40, 42]
	PIO-Tuba Search	16	[1, 4, 7, 8, 13, 14, 15, 16, 25, 27, 28, 29, 32, 36, 37, 39]

Table 10
The Selected features from BoT-IoT by LS-PIO based on different fitness function and local search.

Approach	Local search	Number of features	Selected features
iForest	PIO-Hill-Climbing	4	[3, 11, 12, 18]
	PIO-Tuba Search	3	[3, 11, 15]
OC-SVM	PIO-Hill-Climbing	5	[3, 4, 9, 11, 17]
	PIO-Tuba Search	4	[2, 3, 11,12]
LOF	PIO-Hill-Climbing	4	[3, 11, 13,14]
	PIO-Tuba Search	4	[3, 4, 11, 17]

5.4.1. LS-PIO results

Tables 7–10 show the results of evaluating LS-PIO feature selection algorithm using $KDDCUP^{99}$, NSL-KDD, UNSW-NB15 and BoT-IoT datasets, respectively. Various one-class classifier fitness functions, i.e. iForest, OC-SVM, and LOF, have been considered to compare between PIO-Hill Climbing and PIO-Tabu Search local search algorithm for feature selection. As an output of this evaluation, we got the number of selected features and the index of the selected features for each dataset.

One salient result is that PIO-Tabu feature selection outperforms over Hill Climbing algorithm in terms of the number of selected features in all datasets. Moreover, OC-SVM classifier has presented the highest number of features for both versions of the local search and for all datasets. Fig. 6, Figs. 7 and 8 show the number of selected features for Hill Climbing and Tabu Search when OC-SVM, iForest and LOT classifiers are used on $KDDCUP^{99}$, NSL-KDD and UNSW-NB15 respectively. Based on this result we adopt PIO-Tabu for feature selection when we have conducted the rest of experiments in this paper.

Table 11 presents the evaluation of LS-PIO feature selection for the four datasets based on the selected set of features presented in Tables 7–10 in terms of TPR% and FPR%. The results show that Tabu search algorithm has achieved higher TPR and lower FPR compared with Hill Climbing algorithm in all different classifiers and all datasets. On the other hand, iForest classifier has achieved higher TPR compared with OC-SVM and LOF in $KDDCUP^{99}$ and BoT-IoT datasets. Although, OC-SVM and LOF classifiers have achieved higher TPR compared with iForest classifier in NSL-KDD and UNSW-NB15 but they have also result in higher FPR. When comparing OC-SVM and LOF, we can notice that results vary based on the dataset used. We have used results of this experiment to set the weight for the weighted majority voting, performed in ensemble learning, to give different weights to results that come from various classifiers. For example the highest weight was assigned to iForest classifier.

Table 12 shows a comparison between LS-PIO and other rival recently proposed feature selection algorithms. The comparison has been conducted in terms of the number of features, accuracy, F-score and AUC. A "–" in the table indicates a non-applicable/not-available value. It can be noticed that our proposed feature selection algorithm, LS-PIO outperforms FMIFS (Ambusaidi et al., 2016) in terms of the number of selected features and the accuracy while the same number of selected features have been reached in Li et al. (2018) and LS-PIO outperforms DPC in terms of accuracy. Moreover, the comparison results show that LS-PIO outperforms (Alazzam et al., 2020) in terms of the accuracy considering NSL-KDD and UNSW-NB15 datasets. When comparing LS-PIO with REPT that was proposed by Tama et al. (2019) then we can notice that LS-PIO is able to come up with less number of selected features for NSL-KDD and UNSW-NB15 datasets and better accuracy with the later one. However with NSL-KDD the accuracy of Tama et al. (2019) is better than LS-PIO. F-Score value for this method has not been specified for the above mentioned datasets.

In Khraisat et al. (2020) C5-DT has been used as a feature selection with a OC-SVM as a wrapper function which has achieved an F-score of 83.24% compared to better 89.1% F-score that has been achieved by LS-PIO. Similar result has been concluded when comparing LS-PIO with Pérez et al. (2019) where the accuracy of LS-PIO is apparently higher with higher F-Score values. Since BoT-IoT is a recent dataset, we were unable to find one class classifier based feature selection other than (Kareem et al., 2022) to compare with. Results of this comparison shows that out approach accuracy is slightly better than the GTO-BSA feature selection method proposed by Kareem et al. (2022).

Table 11
The evolutions for LS-PIO in different datasets based on the selected set of features for Datasets in terms of TPR % and FPR %.

Approach	$KDDCUP^{99}$				NSL-KDD				UNSW-NB15				BoT-IoT			
	Hill Climbing		Tabu Search		Hill Climbing		Tabu Search		Hill Climbing		Tabu Search		Hill Climbing		Tabu Search	
	TPR %	FPR %	TPR %	FPR %	TPR %	FPR %	TPR %	FPR %	TPR %	FPR %	TPR %	FPR %	TPR %	FPR %	TPR %	FPR %
iForest	99.3	5.1	99.9	4.3	92.7	13.9	93.2	13.4	95.3	16.5	97.8	14.3	98.7	3.2	99.2	1.5
OC-SVM	97.8	3.1	98.1	2.1	96.5	27.6	98.3	29.3	96.8	43.1	98.4	45.8	98.3	3.5	98.94	1.9
LOF	99.0	15.3	99.7	14.3	90.9	20.3	95.6	21.3	96.8	30.2	97.8	32.1	97.25	4.2	98.2	2.75

Table 12
Comparison between LS-PIO and other feature selection algorithms.

Dataset	Reference	Feature selection	Classifier	Num.Fet	Accuracy %	F-score %	AUC
$KDDCUP^{99}$	Ambusaidi et al. (2016)	FMIFS	LS-SVM	19	99.70	–	–
	Li et al. (2018)	DPC	DPNN	15	96.50	–	–
	Proposed model	LS-PIO	iForest	15	99.82	97.23	96.32
NSL-KDD	Alazzam et al. (2021)	PIO	OC-SVM	–	83	–	–
	Khraisat et al. (2020)	C5-DT	OC-SVM	–	–	83.24	75.2
	Tama et al. (2019)	REPT	PSO	17	96.38	–	–
	Proposed model	LS-PIO	iForest	10	94.7	89.1	87.63
UNSW-NB15	Alazzam et al. (2021)	PIO	OC-SVM	–	52.9	–	–
	Pérez et al. (2019)	Encoder	LoF	–	82	82	–
	Tama et al. (2019)	REPT	PSO	9	81.53	–	–
	Proposed model	LS-PIO	iForest	8	94.45	91.35	89.52
BoT-IoT	Kareem et al. (2022)	GTO-BSA	–	–	96.22	–	–
	Proposed model	LS-PIO	iForest	3	97.37	94.88	95.68

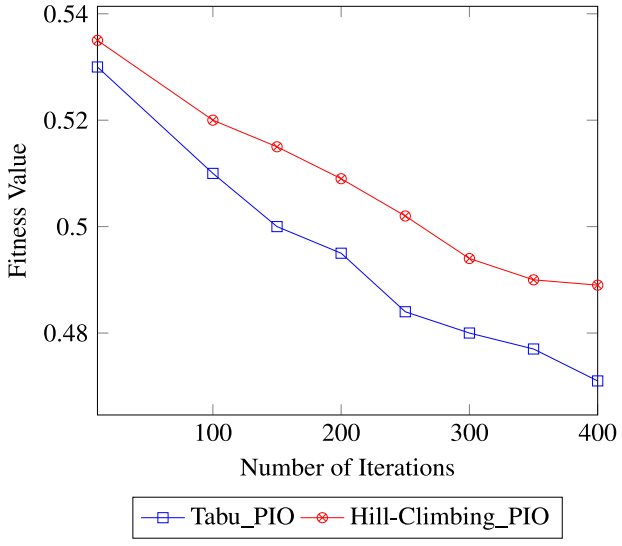

Fig. 9. Tabu and Hill Climbing Convergence curve on $KDDCUP^{99}$.

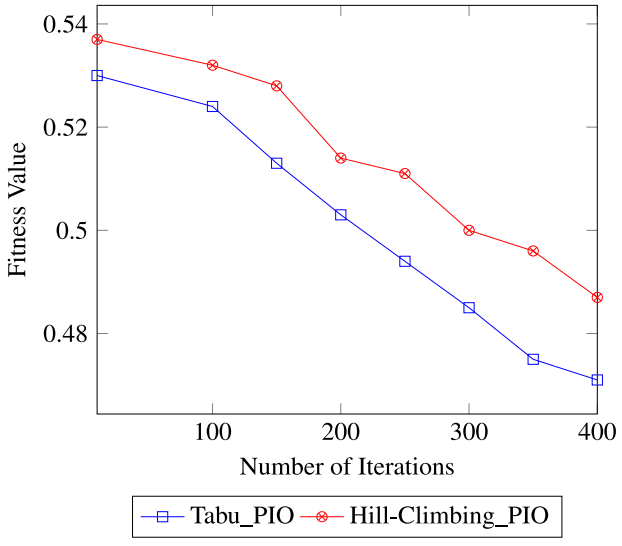

Fig. 10. Tabu and Hill Climbing Convergence curve on NSL-KDD.

To summarize results of Table 12 we can notice that LS-PIO has achieved better accuracy than all other method except REPT which is slightly higher. The F-Score of REPT has not been calculated which has its role in this comparison especially with such a minor difference in accuracy. Additionally, NSL-KDD is not an IoT related dataset, but due to the limitation of one class classifiers available for BoT-IoT we have included NSL-KDD to enrich the evaluation process. In terms of AUC, which reflect the measure of separability, we can notice values in the high eighty's for NSL-KDD and UNSW-NB15 datasets which indicates excellent level of separation between classes. For BoT-IoT the value, i.e. 95.68, of AUC indicates outstanding separability level.

In the $KDDCUP^{99}$dataset, In Fig. 9 the convergence curve for LS-PIO for feature selection algorithms is shown. The result shows that Tabu-PIO has faster convergence than Hill Climbing-PIO since it shows more enhancement to reach the minimum fitness value as the number of iterations increases. It can be noticed that tabu-PIO improves considerable decrease in the first 50 iterations and continues to improve the solution's quality, whereas Hill-PIO improves exponential decrease in the first 80 iterations.

Fig. 10 illustrates the convergence curve for both LS-PIO algorithm of feature selection for NSL-KDD dataset. Tabu-PIO is again able to achieve faster convergence than Hill Climbing-PIO. Moreover, Tabu is able to improve its performance as the number of iteration increases as the case with the previous result. It can be noticed that tabu-PIO shows substantial enhancement after the first 100 iterations and keeps enhancing the quality of the solution, while the Hill-PIO shows the same level of enhancement after the first 150 iterations.

Fig. 11 illustrates Tabu and Hill Climbing Convergence curve in LS-PIO of feature selection algorithm for UNSW-NB15. The result shows the ability of Tabu-PIO to perform faster in terms of convergence than Hill Climbing-PIO due to enhancing the fitness function as the number of iterations increases until it reaches the minimum fitness value. We have noticed that tabu-PIO shows a noticeable enhancement after the first 97 iterations and continues enhancing the quality of the

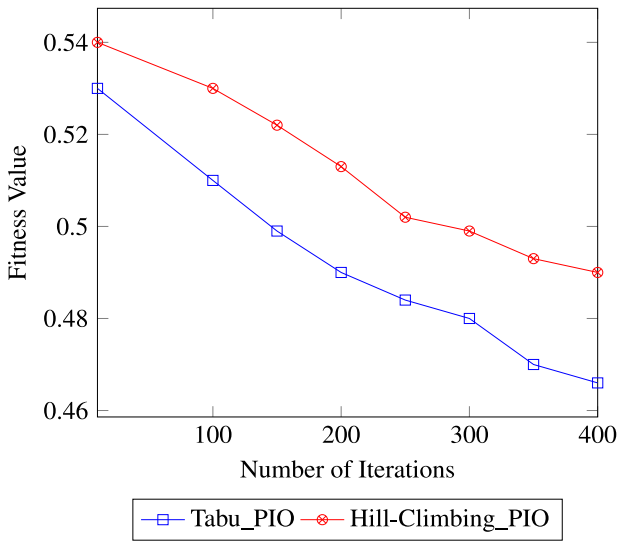

Fig. 11. Tabu and Hill Climbing Convergence curve on UNSW-NB15.

solution, whereas the Hill-PIO shows that enhancement after the first 142 iterations.

5.4.2. NIDS results

In Table 13 one can notice the overfitting of results when comparing the Test.Acc% and Tr.Acc% with the resulted high variance. However, even with this overfitting case, One can notice that the introduction of the ensemble technique allows the achievement of better results compared to the rival techniques for all datasets.

Table 14 illustrates the evaluation of the various NIDS schemes fed with the selected features resulting from Tabu-PIO feature selection algorithm. The use of Tabu-PIO feature selection algorithm has improved the performance of iForest, OC-SVM, LOF based NIDS and even the Ensemble based NIDS.

One important improvement is the high reduction of overfitting in the results when comparing accuracies in Tables 13 and 14. This indicates that our LS-PIO approach helps not only in getting higher accuracy but also in achieving good fitting in the obtained results.

Table 13 presents the evaluation of the various considered NIDS without the proposed feature selection algorithm, i.e using all the available features. The performance of various techniques used with the four different datasets. The ensemble approach can be compared with iForest, OC-SVM, and LOF. In this experiment, 10-folds cross validation has been applied to validate the results of NIDS methods in terms of training accuracy (Tr.Acc%), and validating accuracy (Val.Acc%). The considered performance metrics are the TPR%, FPR%, three accuracy values, i.e. training accuracy (Tr.Acc%), validating accuracy (Val.Acc%) and testing accuracy (Test.Acc%), F-score% and Tr.Acc% Val.Acc% and AUC.

Table 15 presents comparison between our LS-PIO based ensemble approach for NIDS and other rival ensemble approaches found in the literature. The comparison has been done in terms accuracy, F-Score and AUC for all datasets. Results of AUC indicated excellent and outstanding level of separability of classes for the proposed ensemble method. Furthermore, the propose ensemble method achieves higher accuracy than all other approaches and considering all datasets. In terms of F-score, results show that the proposed ensemble approach achieves better values than all other approaches except for the one proposed in Khraisat et al. (2020) for the $KDDCUP^{99}$ dataset which is not an IoT dataset that has been included to enrich the evaluation and the comparison process in this paper.

Chauhan and Atulkar (2021) tree-based ensemble methods used in this work for identifying the attacks in the IoT network are Random Forest, LGBM, Extra Tree, Gradient Boost, XGBoost.

5.5. Statistical analysis

Statistics provides a powerful procedures to test the significance of differences between multiple methods. This section presents an analysis of the experimental results in order to conduct a significance statistical analysis between LS-PIO model and the original PIO model (Alazzam et al., 2020).

To check the normality distribution, we use Shapiro–Wilk test with 0.05 critical levels to test the distribution of the results (see Table 16. To make a fair test, we randomly select 20 results obtained by our approach to testing the normality distribution of our results.

In order to determine the statistically significant for the observed outcome sample data, we have conducted a T-test in order to measure the p-value of the test statistics. P-value represents the level of marginal significance considered for a statistical hypothesis test. It represents the probability of the occurrence of a certain event. Additionally, we have selected the significance level (alpha) to be 0.05. The smaller p-value, the observed outcome is more statistically significant (Rice, 1989).

The statistical analysis of the results has been conducted for LS-PIO and PIO models and considering two datasets, i.e. UNSW-NB15 and BoT-IoT. Table 16 shows the Mean, Std.Dev, Std.Err, and the P-Value or the significance level (sig.). The P value is less than the critical level 0.05 which indicates that the results are statistically significant.

6. Conclusion

In this paper, a new feature selection algorithm based on a modified version of PIO named as LS-PIO has been proposed. Two local search algorithms which are Tabu and Hill Climbing, have been investigated to enhance the performance of PIO. LS-PIO feature selection algorithm is designed to decrease the number of features needed to build robust NIDS. Tabu-PIO algorithm showed better results compared with Hill climbing-PIO algorithm in terms of TPR and FPR. Moreover, it presents less number of selected features in all datasets.

Tabu-PIO feature selection algorithm lessened the number of selected features for UNSW-NB15, $KDDCUP^{99}$ and NSL-KKD from 49 to 8, from 42 to 12 and from 42 to 10 features only for the selected datasets respectively. Moreover, in Bot-IoT it reduces the number of features to three selected features only.

NIDS was built based on ensemble learning that uses only one class classifiers which are LOF, OC-SVM and iForest. The proposed NIDS takes advantages of the selected features from LS-PIO to enhance the results to detect the anomalies compared with each one-class classifier that has been used individually.

Results show that LS-PIO helps in achieving higher accuracy and better good fitting of the results. LS-PIO has been found to be lightweight since K-Means has been used as pre-processing stage to reduce the running time of the method which makes NIDS based on LS-PIO suitable for IoT environment.

CRediT authorship contribution statement

Orieb Abu Alghanam: Conceptualization, Methodology, Software, Validation, Writing and editing. **Wesam Almobaideen:** Investigation, Writing – original draft, Writing – review & editing, Supervision. **Maha Saadeh:** Supervision, Writing – review & editing. **Omar Adwan:** Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

the datasets are benchmark dataset so it is available.

Table 13
The evaluation of the proposed NIDS using several dataset using all features.

Dataset	Technique	TPR %	FPR %	Tr.Acc %	Val.Acc %	Test Acc %	F-score %	AUC
$KDDCUPP^{99}$	iForest	44.2	14.32	75.3	73.64	53.13	51.63	49.36
	OC-SVM	35.7	25.25	46.87	42.98	35.01	33.64	31.25
	LOF	40.2	21.17	63.87	59.48	42.74	39.09	46.97
	Ensemble approach	**47.28**	**13.20**	**77.13**	**75.83**	**54.83**	**52.41**	**47.12**
NSL-KDD	iForest	39.3	10.9	64.87	61.58	50.27	49.93	46.82
	OC-SVM	29.2	23.5	37.25	34.28	24.7	20.37	21.36
	LOF	37.3	25.3	43.92	41.65	35.32	33.37	32.58
	Ensemble approach	**40.32**	**9.08**	**74.49**	**71.93**	**51.43**	**52.77**	**48.97**
UNSW-NB15	iForest	13.7	7.3	58.69	55.21	40.9	37.35	36.24
	OC-SVM	17.3	37.3	32.56	29.73	22.01	20.64	19.69
	LOF	35.2	29.17	59.31	56.98	42.9	41.58	38.65
	Ensemble approach	**27.51**	**6.19**	**61.65**	**59.18**	**41.53**	**39.74**	**37.87**
BoT-IoT	iForest	50.67	14.17	71.3	69.56	65.51	63.25	74.20
	OC-SVM	42.84	29.12	61.87	60.41	41.01	50.36	61.26
	LOF	45.07	17.29	65.3	69.56	65.51	63.25	64.32
	Ensemble approach	**52.3**	**15.1**	**72.91**	**70.34**	**69.32**	**70.79**	**79.84**

Table 14
The evaluation of the proposed NIDS using Several dataset based on Tabu Search PIO Features.

Dataset	Technique	TPR	FPR	Tr.Acc %	Val.Acc %	Test Acc %	F-score	AUC
$KDDCUPP^{99}$	iForest	99.9	4.8	99.98	99.32	99.10	95.30	91.21
	OC-SVM	98.4	2.5	99.57	98.24	96.97	93.99	88.36
	LOF	98.4	31.7	99.35	99.27	98.40	89.20	90.32
	LS-PIO based ensemble	**99.7**	**2.4**	**99.3**	**99.01**	**98.90**	**95.89**	**90.46**
NSL-KDD	iForest	92.7	13.2	96.58	95.3	91.2	89.79	87.69
	OC-SVM	99.8	28.6	91.35	90.61	85.7	84.3	82.35
	LOF	95.9	23.1	90.05	88.32	84.98	83.5	80.89
	LS-PIO based ensemble	**97.5**	**11.2**	**98.43**	**97.95**	**96.93**	**91.9**	**87.19**
UNSW-NB15	iForest	96.6	19.7	95.65	93.4	90.1	89.5	84.20
	OC-SVM	98.4	45.8	73.25	70.94	62.9	72.2	79.89
	LOF	97.8	32.1	84.32	82.97	78.3	83.9	81.94
	LS-PIO based ensemble	**96.7**	**11.1**	**97.94**	**96.52**	**93.3**	**90.7**	**86.58**
BoT-IoT	iForest	99.16	1.7	99.8	99.7	97.2	94.35	96.20
	OC-SVM	98.4	32.8	82.36	80.97	77.79	76.28	74.79
	LOF	98.78	19.81	92.48	91.27	88.93	87.89	85.85
	LS-PIO based ensemble	**99.7**	**1.02**	**99.94**	**98.56**	**98.73**	**96.7**	**97.98**

Table 15
Comparisons of the Proposed approach and other related ensemble approaches for NIDS.

Dataset	Reference	Technique	Acc %	F-score	AUC
$KDDCUPP^{99}$	Kaplan and Alptekin (2020)	iForest +LoF+OC-SVM	89.5	90.8	–
	Khraisat et al. (2020)	LoF+OC-SVM	–	98.4	–
	The proposed approach	iForest +LoF+OC-SVM	98.90	95.89	90.46
NSL-KDD	Tama et al. (2019)	Rotation forest+Bagging	96.38	–	–
	Jain and Kaur (2021)	LR+RF	78.9	90.0	–
	The proposed approach	iForest +LoF+OC-SVM	96.93	91.90	87.19
UNSW-NB15	Tama et al. (2019)	Rotation forest+Bagging	81.53	–	–
	Rashid et al. (2020)	Bagging	82.63	81.00	–
	The proposed approach	iForest +LoF+OC-SVM	93.30	90.70	86.58
BoT-IoT	Khraisat et al. (2019)	C4+OCSVM	92.50	–	–
	The proposed approach	iForest +LoF+OC-SVM	98.73	96.7	97.98

Table 16
Statistical analysis.

Dataset	Metric	LS-PIO			PIO			P-Value
		Mean	Std.Dev	Std.Err	Mean	Std.Dev	Std.Err	
UNSW-NB15	F-Score	90.4759	1.2655	0.28298	79.8635	1.04927	0.23462	0.001
UNSW-NB15	Acc	94.2952	0.92529	0.20192	81.0071	1.61397	0.35220	0.001
BoT-IoT	F-Score	91.9360	1.66102	0.37142	81.5325	1.36147	0.30443	0.001
BoT-IoT	Acc	95.4735	1.45166	0.32460	81.7120	2.29576	0.51335	0.001

References

Abu, M., Rahayu, S., Ariffin, D. A., & Robiah, Y. (2018). Cyber threat intelligence – issue and challenges. *Indonesian Journal of Electrical Engineering and Computer Science*, *10*, 371–379.

AbuAlghanam, O., Albdour, L., & Adwan, O. (2021). Multimodal biometric fusion online handwritten signature verification using neural network and support vector machine. *Transactions*, *7*, 8.

Abualghanam, O., Qatawneh, M., & Almobaideen, W. (2019). A survey of key distribution in the context of internet of things. *Journal of Theoretical and Applied Information Technology*, *97*(22), 3217–3241.

AbuAlghanam, O., Qatawneh, M., Almobaideen, W., & Saadeh, M. (2022). A new hierarchical architecture and protocol for key distribution in the context of IoT-based smart cities. *Journal of Information Security and Applications*, *67*, Article 103173.

Aguilar, D. L., Loyola-González, O., Medina-Pérez, M. A., Cañete-Sifuentes, L., & Choo, K.-K. R. (2021). PBC4occ: A novel contrast pattern-based classifier for one-class classification. *Future Generation Computer Systems*, *125*, 71–90.

Alazzam, H., Sharieh, A., & Sabri, K. E. (2020). A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. *Expert Systems with Applications*, *148*, Article 113249.

Alazzam, H., Sharieh, A., & Sabri, K. E. (2021). Lightweight intelligent network intrusion detection system using OCSVM and pigeon inspired optimizer. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*.

Alghushairy, O., Alsini, R., Soule, T., & Ma, X. (2021). A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing*, *5*(1), 1.

Alhajjar, E., Maxwell, P., & Bastian, N. (2021). Adversarial machine learning in network intrusion detection systems. *Expert Systems with Applications*, Article 115782.

Almobaideen, W., & Altarawneh, M. (2020). Fog computing: survey on decoy information technology. *International Journal of Security and Networks*, *15*(2), 111–121.

Alsahaf, A., Petkov, N., Shenoy, V., & Azzopardi, G. (2021). A framework for feature selection through boosting. *Expert Systems with Applications*, Article 115895.

Alzaqebah, M., Jawarneh, S., Mohammad, R. M. A., Alsmadi, M. K., Al-marashdeh, I., Ahmed, E. A. E., Alrefai, N., & Alghamdi, F. A. (2021). Hybrid feature selection method based on particle swarm optimization and adaptive local search method. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*(3), 2414–2422.

Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Transactions on Computers*, *65*(10), 2986–2998.

Asassfeh, M., Obeid, N., & Almobaideen, W. (2020). Anonymous authentication protocols for IoT based-healthcare systems: A survey. *International Journal of Communication Networks and Information Security*, *12*(3), 302–315.

Aydın, H., Orman, Z., & Aydın, M. A. (2022). A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment. *Computers & Security*, Article 102725.

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*: *Tech. Rep.*, Irvine, CA, USA: School Inf. Comput. Sci., Univ. California Irvine.

Bouzoubaa, K., Nsiri, B., & Taher, Y. (2021). Predicting DOS-DDOS attacks: Review and evaluation study of feature selection methods based on wrapper process. *(IJACSA) International Journal of Advanced Computer Science and Applications*, *12*(5), 131–145.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Camiña, J. B., Medina-Pérez, M. A., Monroy, R., Loyola-González, O., Villanueva, L. A. P., & Gurrola, L. C. G. (2019). Bagging-RandomMiner: A one-class classifier for file access-based masquerade detection. *Machine Vision and Applications*, *30*(5), 959–974.

Carletti, M., Terzi, M., & Susto, G. A. (2020). Interpretable anomaly detection with diffi: Depth-based feature importance for the isolation forest. arXiv preprint arXiv:2007.11117.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58.

Chauhan, P., & Atulkar, M. (2021). Selection of tree based ensemble classifier for detecting network attacks in IoT. In *2021 international conference on emerging smart computing and informatics (ESCI)* (pp. 770–775). IEEE.

Chen, B., Lei, H., Shen, H., Liu, Y., & Lu, Y. (2019). A hybrid quantum-based PIO algorithm for global numerical optimization. *Science China. Information Sciences*, *62*(7), 1–12.

Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems* (pp. 161–168).

da Costa, N. L., de Lima, M. D., & Barbosa, R. (2021). Evaluation of feature selection methods based on artificial neural network weights. *Expert Systems with Applications*, *168*, Article 114312.

Duan, H., & Qiao, P. (2014). Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *International Journal of Intelligent Computing and Cybernetics*.

Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, *42*, 2670–2679.

Feng, J., & Gong, Z. (2022). A novel feature selection method with neighborhood rough set and improved particle swarm optimization. *IEEE Access*, *10*, 33301–33312.

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, *50*, Article 102419.

Gopalan, S., Raza, A., & Almobaideen, W. (2021). IoT security in healthcare using AI: A survey. In *2020 international conference on communications, signal processing, and their applications (ICCSPA)* (pp. 1–6). IEEE.

Guilford, T., Roberts, S., Biro, D., & Rezek, I. (2004). Positional entropy during pigeon homing II: navigational interpretation of Bayesian latent state models. *Journal of Theoretical Biology*, *227*(1), 25–38.

Imrana, Y., Xiang, Y., Ali, L., & Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. *Expert Systems with Applications*, *185*, Article 115524.

Jain, M., & Kaur, G. (2021). Distributed anomaly detection using concept drift detection based hybrid ensemble techniques in streamed network data. *Cluster Computing*, *24*(3), 2099–2114.

Kang, S.-H., & Kim, K. J. (2016). A feature selection approach to find optimal feature subsets for the network intrusion detection system. *Cluster Computing*, *19*, 325–333.

Kaplan, M. O., & Alptekin, S. E. (2020). An improved bigan based approach for anomaly detection. *Procedia Computer Science*, *176*, 185–194.

Kareem, S. S., Mostafa, R. R., Hashim, F. A., & El-Bakry, H. M. (2022). An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection. *Sensors*, *22*(4), 1396.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. *Electronics*, *8*(11), 1210.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2020). Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine. *Electronics*, *9*(1), 173.

Kittidachanan, K., Minsan, W., Pornnopparath, D., & Taninpong, P. (2020). Anomaly detection based on GS-OCSVM classification. In *2020 12th international conference on knowledge and smart technology (KST)* (pp. 64–69). IEEE.

Koroniotis, N. (2020). *Designing an effective network forensic framework for the investigation of botnets in the Internet of Things* (Ph.D. thesis), Sydney, Australia: University of New South Wales.

Koroniotis, N., & Moustafa, N. (2020). Enhancing network forensics with particle swarm and deep learning: The particle deep framework. arXiv preprint arXiv:2005.00722.

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, *100*, 779–796.

Li, L., Zhang, H., Peng, H., & Yang, Y. (2018). Nearest neighbors based density peaks approach to intrusion detection. *Chaos, Solitons & Fractals*, *110*, 33–40.

Lippmann, R. P., Graf, I., Wyschogrod, D., Webster, S. E., Weber, D. J., & Gorton, S. (1998). The 1998 DARPA/AFRL off-line intrusion detection evaluation. In *First international workshop on recent advances in intrusion detection (RAID)* (pp. 163–181).

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth Ieee international conference on data mining* (pp. 413–422). IEEE.

Ma, T., Zhou, H., Jia, D., Al-Dhelaan, A., Al-Dhelaan, M., & Tian, Y. (2019). Feature selection with a local search strategy based on the forest optimization algorithm. *Computer Modeling in Engineering & Sciences (CMES)*, *121*(2), 569–592.

Maglaras, L. A., & Jiang, J. (2014). Ocsvm model combined with k-means recursive clustering for intrusion detection in scada systems. In *10th international conference on heterogeneous networking for quality, reliability, security and robustness* (pp. 133–134). IEEE.

Maglaras, L. A., Jiang, J., & Cruz, T. J. (2016). Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems. *Journal of Information Security and Applications*, *30*, 15–26.

Mbanaso, U. M., & Dandaura, E. (2015). The cyberspace: Redefining a new world. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *17*(3), 17–24.

Medina-Pérez, M. A., Monroy, R., Camiña, J. B., & García-Borroto, M. (2017). Bagging-tpminer: A classifier ensemble for masquerader detection based on typical objects. *Soft Computing*, *21*(3), 557–569.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 military communications and information systems conference (MilCIS)* (pp. 1–6). IEEE.

Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, *25*(1–3), 18–31.

Naseri, T. S., & Gharehchopogh, F. S. (2022). A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. *Journal of Network and Systems Management*, *30*(3), 1–27.

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462.

Paulauskas, N., & Bagdonas, Ą. F. (2015). Local outlier factor use for the network flow anomalydetection. *Security and Communication Networks*, *8*, 4203–4212.

Pérez, D., Alonso, S., Morán, A., Prada, M. A., Fuertes, J. J., & Domínguez, M. (2019). Comparison of network intrusion detection performance using feature representation. In *International conference on engineering applications of neural networks* (pp. 463–475). Springer.

Pérez, D., Alonso, S., Morán, A., Prada, M. A., Fuertes, J. J., & Domínguez, M. (2021). Evaluation of feature learning for anomaly detection in network traffic. *Evolving Systems*, *12*(1), 79–90.

Qatawneh, M., Almobaideen, W., & AbuAlghanam, O. (2020). Challenges of blockchain technology in context internet of things: A survey. *International Journal of Computer Applications*, *975*, 8887.

Rajasegarar, S., Leckie, C., & Palaniswami, M. (2008). CESVM: Centered hyperellipsoidal support vector machine based anomaly detection. In *2008 IEEE international conference on communications* (pp. 1610–1614). IEEE.

Rashid, M. M., Kamruzzaman, J., Hassan, M. M., Imam, T., & Gordon, S. (2020). Cyberattacks detection in iot-based smart city applications using machine learning techniques. *International Journal of Environmental Research and Public Health*, *17*(24), 9347.

Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)*, *2*(12), 1848–1853.

Rice, W. R. (1989). Analyzing tables of statistical tests. *Evolution*, *43*(1), 223–225.

Sohn, I. (2020). Deep belief network based intrusion detection techniques: A survey. *Expert Systems with Applications*, Article 114170.

Sun, H., & Duan, H. (2014). PID controller design based on prey-predator pigeon-inspired optimization algorithm. In *2014 IEEE international conference on mechatronics and automation* (pp. 1416–1421). IEEE.

Tama, B. A., Comuzzi, M., & Rhee, K.-H. (2019). TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. *IEEE Access*, *7*, 94497–94507.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1–6). IEEE.

Tian, Y., Mirzabagheri, M., Bamakan, S. M. H., Wang, H., & Qu, Q. (2018). Ramp loss one-class support vector machine; a robust and effective approach to anomaly detection problems. *Neurocomputing*, *310*, 223–235.

Vijayanand, R., & Devaraj, D. (2020). A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network. *IEEE Access*, *8*, 56847–56854.

Wan, Y., Wang, M., Ye, Z., & Lai, X. (2016). A feature selection method based on modified binary coded ant colony optimization algorithm. *Applied Soft Computing*, *49*, 248–258.

Wu, Q., Ma, Z., Fan, J., Xu, G., & Shen, Y. (2019). A feature selection method based on hybrid improved binary quantum particle swarm optimization. *IEEE Access*, *7*, 80588–80601.

Xiong, Y., & Zuo, R. (2020). Recognizing multivariate geochemical anomalies for mineral exploration by combining deep learning and one-class support vector machine. *Computers & Geosciences*, *140*, Article 104484.

Zhou, Y., Ren, H., Li, Z., & Pedrycz, W. (2022). Anomaly detection based on a granular Markov model. *Expert Systems with Applications*, *187*, Article 115744.

Zhou, H., Zhang, J., Zhou, Y., Guo, X., & Ma, Y. (2021). A feature selection algorithm of decision tree based on feature weight. *Expert Systems with Applications*, *164*, Article 113842.