Feature Selection using Improved Nomadic People Optimizer in Intrusion Detection

Zinah Sattar Jabbar Aboud
The Lebanese University, Lebanon | Department of Computer Technology Engineering, College of Information Technology, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq
sattarzeina@gmail.com (corresponding author)

Rami Tawil
The Lebanese University, Lebanon
rami.tawil@ul.edu.lb

Mustafa Salam Kadhm
Computer Department, College of Basic Education, Mustansiriyah University, Iraq
mst.salam@uomustansiriyah.edu.iq

Received: 16 September 2024 | Revised: 8 October 2024 and 10 October 2024 | Accepted: 12 October 2024

DOI: https://doi.org/10.48084/etasr.9020

ABSTRACT

Intrusion Detection (ID) in network communication and Wireless Sensor Networks (WSN) is a big challenge that has grown with the rapid development of these technologies. Various types of intrusion attacks may occur to the transferred data of such networks and various ID methods and algorithms have been proposed. One powerful tool used in this field is Machine Learning (ML), which has achieved satisfied detection results. However, these results with the available ID datasets can be further improved. This paper proposes an accurate approach for ID in the network and WSN using ML methods including chaotic map, Nomadic People Optimizer (NPO), and Support Vector Machine (SVM). The proposed approach has five main stages which are: data collection, pre-processing, feature selection, classification, and evaluation. An improved version of NPO based on chaotic map and Cauchy mutation called CNPO is proposed. The proposed scheme uses chaotic maps to initialize the population and Cauchy mutation for solution distribution. Besides, the proposed fitness function based on SVM is proposed. The CNPO is employed for the feature selection task. The proposed approach was evaluated in two datasets, NSL-KDD, and WSN-DS, with accuracy of 99.97% and 99.99, respectively.

Keywords-IDS; NPO; CNPO; chaos; Cauchy; SVM; classification; feature selection

I. INTRODUCTION

With the rapid development of network and communication fields, security challenges such as Intrusion Detection (ID) have emerged. The internet poses a great thread on the user privacy and data security [1]. Users on internet suffer from several types of cyber-attacks [2]. In order to prevent all possible attacks in the internet, a system capable of detecting anomalies and protecting the network called Intrusion Detection Systems (IDS) is required [3-6]. IDS is a critical part of cybersecurity, designed to detect unauthorized access in a computer network or a computer system. The IDS monitors the network traffic and system activities to identify any possible threats such as: suspicious behavior, insider threats, and malware. When IDS detects unauthorized activities, the system will generate alerts to quickly respond, mitigate, and prevent other attacks [7-9]. Feature selection is used in IDS by identifying the most relevant features from the considered dataset that contribute to accurate ID. Feature selection improves the IDS performance, efficiency, and interpretability by selecting only the most informative features from the available data. In general, there are four types of feature selection techniques for IDS: filters, wrappers, embedded selectors, and dimensionality reduction [10-12]. Over the years, various feature selection methods have been proposed and established [13-17]. Authors in [18, 19] used cuckoo algorithm and improved cuckoo algorithm for the feature selection task. In [20], the Modified Metaheuristics with Weighted Majority Voting Ensemble Deep Learning (MM-WMVEDL) model for IDS was proposed, while in [21], Explorated Particle Swarm Optimization (PSO) centred Sea Turtle Foraging Algorithm (EXPSO-STFA) was employed for feature selection. Other optimization algorithms [22-25] have been applied successfully in IDS feature selection and satisfying detection results in various IDS datasets have been obtained.

Nomadic People Optimizer (NPO) is a recent optimization algorithm [26]. NPO is a metaheuristic algorithm inspired by the life style of nomadic communities in the desert. NPO is designed based on the multi-swarm approach utilizing several clans for finding the best solution via following the leader position. NPO has been applied in several researches and achieved optimum optimization results [27-28].

Authors in [29] present a comprehensive analysis of IDSs using several Machine Learning (ML) algorithms, such as Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF). The work aims to evaluate the performances of the algorithms in the NSL-WSN dataset in various scenarios. SVM achieved the highest accuracy of 95.2%, with 92.6% precision, 94.3% recall, and 93.4% F1-score. Authors in [30] propose an algorithm for real-time IDS called PACENIDS using an ensemble of Altered Bi-directional Long Short-Term Memory (ABILSTM) and Customized Bi-directional Gated Recurrent Unit (CBIGRU). The proposed algorithm was employed to detect attacks in smart cities networks. In order to improve the performance of the proposed algorithm, the authors used a fuzzy feature selection algorithm. PACENIDS achieved a high classification accuracy of 96.59%, i.e. 94.47% without utilizing the feature selection algorithm and 97.67% with the feature selection algorithm, in the NSL-KDD dataset. Authors in [31] propose a hybrid filter-wrapper feature selection method for an IDS called GBA. The proposed method selects a feature subset from the original features to improve the performance of the system. The filter feature selection is based on the Information Gain (IG) algorithm and the wrapper feature selection is based on the Black Hole (BH) algorithm. The aim of GBA is to improve the accuracy of the IDs by initialize the features for classification using IG by ignoring the zero weighted features. GBA achieved a classification accuracy of 96.96% in the NSL-WS dataset. Authors in [32] present a combination approach between optimization and ML algorithms for network ID called KOMIC IDS. Knapsack Optimization (KO) algorithm with a Mutual Information Gain (MIC) filter was used to select the relevant features from the ID dataset. Then, a new set of features was combined with the selected features. MIC was applied again on the combined features to remove the duplicated features and keep only the highest information gain features. Several ML classifiers were used to evaluate the performance of KOMIC IDS and the best obtained results were 97.14% accuracy, 95.53% precision, 99.46% recall, and 97.46% F1-score in the UNSW-NB15 dataset. Authors in [19] proposed a hybrid approach for network ID based on cuckoo algorithm and perceptron neural network. The proposed approach enhances the accuracy of the existing IDs by 1%. Cuckoo algorithm was employed for selecting a subset of the features from the IDS dataset based on a number of characteristics and the perceptron neural network was applied for feature attribute analysis. The proposed approach was evaluated using the KDD-CUP99 dataset and achieved a detection accuracy of 89.8%, 93.41% precision, 99.13% recall, and 97.7% F1-score. Authors in [33] present a method for selecting features for IDS based on thresholding. The proposed method combines three ML methods which are mutual information, thresholding feature selection, and XGBoost classifier. The dependency between the input features and the target features of the IDS dataset is measured using the mutual information method. In thresholding, the optimal number of the selected features is determined for better classification results. XGBoost was applied to classify the selected features. The method was evaluated in three different IDS datasets (UNSW-NB15, NSL-KDD, and CIC-IDS2017) and achieved accuracy values of 87.63%, 80.51, and 99.89%. Authors in [34] propose an attack detection approach in Wireless Sensor Networks (WSNs). The proposed approach uses Stochastic Machine Learning (SML) based on Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs). Principal Component Analysis (PCA) was applied for the reduction of the WSN dataset dimension. HMMs and GMMs were trained using Expectation-Maximization iterative ML for better classification performance. The achieved accuracy was 94.55% in the WSN-DS dataset. Authors in [35] present an approach that integrates ML techniques with the Synthetic Minority Oversampling Technique Tomek Link (SMOTE-TomekLink) algorithm. The approach enhanced the ID accuracy in WSN dataset by balancing the input data in the dataset, eliminating Tomek links, and collecting minority cases. This approach achieved an accuracy of 99.92% in the WSN-DS dataset.

This paper proposes an accurate IDS using an improved version of NPO algorithm called CNPO. The NPO was improved using logistic and Cauchy mutation for improving the solution search space and create optimal diversity. Besides, a proposed fitness function based on SVM is used in NPO. SVM also is used for attack classification and detection. The CNPO is evaluated using two standard ID datasets, NSL-KDD, and WSN-DS.

II. THE PROPOSED APPROACH

In the proposed ID method, several stages are considered. These stages work together to achieve optimum network ID results. The considered stages include: Data collection, pre-processing, feature selection, classification, and evaluation. The main stages of the proposed work are illustrated in Figure 1.

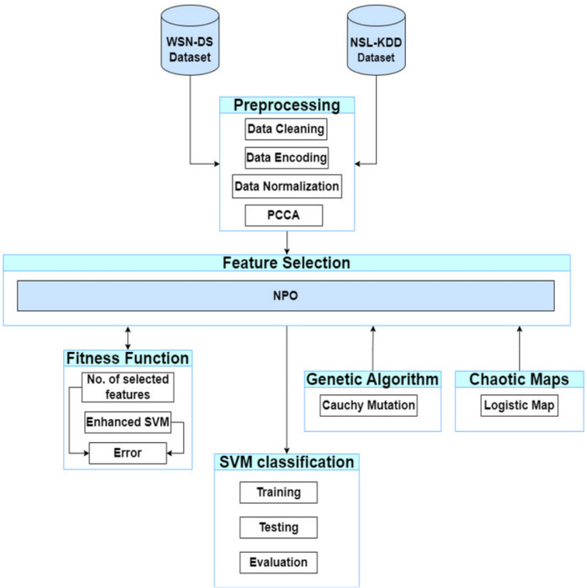

Fig. 1. The proposed approach.

A. Data Collection

Two datasets are used in order to evaluate the performance, NSL-KDD [36] and WSN-DS [37]. The NSL-KDD dataset is an updated version of the KDD'99 dataset.

Each record in the NSL-KDD dataset has 42 attributes (39 numerical and 3 symbolic). Forty one attributes represent data characteristics and the last one class label represents the attack type. The label includes the normal data and four types of attacks which are (denial of service-DoS, Probe, R2L-Remote to user, U2R-User to Root). The dataset has two files, one for training (KDDTrain +) with 125973 records and another for testing (KDDTest +) with 22544 records. Table I shows the number of instances in NSL-KDD dataset.

TABLE I. DETAILS OF THE NSL-KDD DATASET

Attack	Instances	
	Training set	Testing set
Normal	67343	9711
DoS	45927	7456
Probe	11656	2421
R2L	995	2756
U2R	52	200
Total	125973	22544

WSN-DS dataset is built for DoS attack detection. The data of WSN-DS were collected using the LEACH routing protocol. The dataset has 374661 records which represent normal and four types of attacks, namely Blackhole, Grayhole, Flooding, and Scheduling. WSN-DS dataset is described in Table II.

TABLE II. DETAILS OF THE WSN-DS DATASET

Attack	Instances	
	Training set	Testing set
Normal	204174	135892
Blackhole	5999	4050
Grayhole	8653	5943
Flooding	1963	1349
Scheduling	7004	2631
Total	224796	149865

B. Data Pre-processing

In the pre-processing stage, the NSL-KDD and WSN-DS datasets go through data cleaning, encoding, and normalization. The cleaning process is applied to remove the missing and unwanted values such as the values of "num_outbound_cmds" in NSL-KDD that are always equal to zero. Since the datasets have some attributes with text representation (e.g. attack types, protocols, etc.), and the next stages work only with numerical values, an encoding process is applied to convert the text to numerical representation using indexing in the second step of the pre-processing stage. Normal behavior and every attack type will acquire a label to be used in the classification stage. The last step of pre-processing is data normalization. The normalization step reduces the effect of variance of the numerical data range on the classification stage. In the proposed work, the data is normalized into the [0,1] scale using (1) [38]:

$$\mathrm{X_{Scaled}} = \frac{\mathrm{X} - \mathrm{X_{min}}}{\mathrm{X_{max}} - \mathrm{X_{min}}} \qquad (1)$$

Pearson Correlation Coefficient Analysis (PCCA) is a statistical technique used to measure the strength and direction of a linear relationship between two variables. The result is expressed as the Pearson correlation coefficient, denoted as r, which can range from -1 to +1 [39]. The most effective values considered in NSL-KDD and WSN-DS datasets are found by using the PCCA:

$$\mathrm{r}(\mathrm{s_i}, \mathrm{s_j}) = \frac{\mathrm{Cov}\,(\mathrm{s_i}, \mathrm{s_j})}{\sqrt{\mathrm{Var}(\mathrm{s_i})\,.\mathrm{Var}(\mathrm{s_j})}} \qquad (2)$$

where r is the Pearson correlation coefficient, $\mathrm{Var}(\mathrm{s_i})$ and $\mathrm{Var}(\mathrm{s_j})$ are the mean values of two variables, and s_i and s_j are their individual values.

C. Feature Selection

One of the most important stages in the proposed work is feature selection. An efficient feature section stage leads to better detection results. In the proposed work, CNPO is utilized for feature selection process using two chaotic maps (logistic), Cauchy mutation, and a fitness function (based on SVM) is proposed. The CNPO selects the relevant features with highest impact on the obtained results based on optimal feature criteria. The proposed feature selection method is shown in Figure 2.

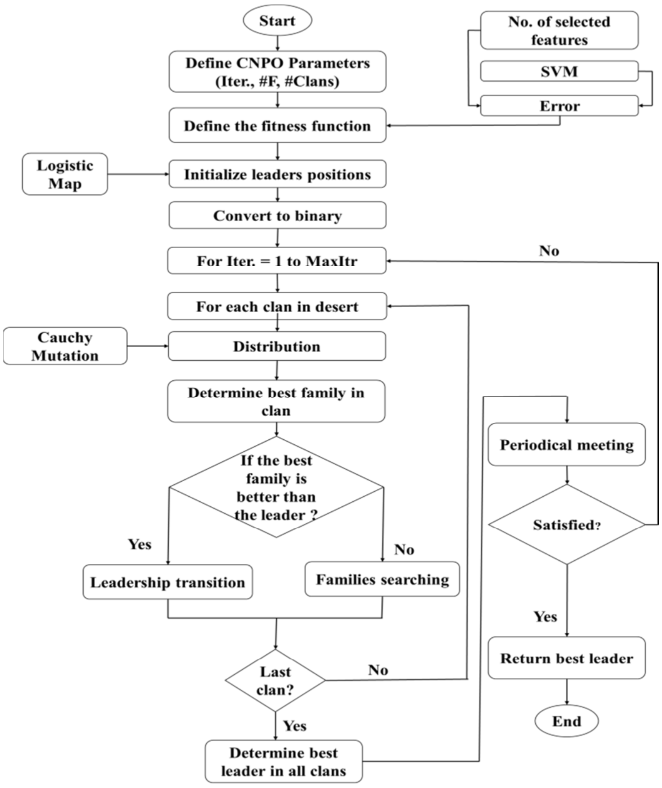

Fig. 2. The proposed feature selection algorithm.

D. Nomadic People Optimizer (NPO) [27]

NPO is a metaheuristic algorithm inspired by the life style of the nomadic communities in the desert searching for useful life sources. NPO is designed based on the multi-swarm approach with several clans searching to find the best solution via following the leader position.

1) NPO Terminology

- Leader (σ): the best local current solution.
- Best Leader (σ^E): the best global current solution (used in the periodical meeting).
- Normal Leader (σ^N): the other leaders (except the best one).
- Family (x): the clan member with lower fitness than the leader.
- Clan (c): leaders with their families (one leader for each family).
- Fitness function (*f*(*x*)): the goodness of position evaluation.
- Direction (Ψ): guide σ^N into σ^E.

2) NPO Algorithm

The NPO algorithm consists of five main steps which are: initialization, semi-circular distribution, family searching, transition of leadership, and periodical meeting. Figure 3 shows the flowchart of the NPO.

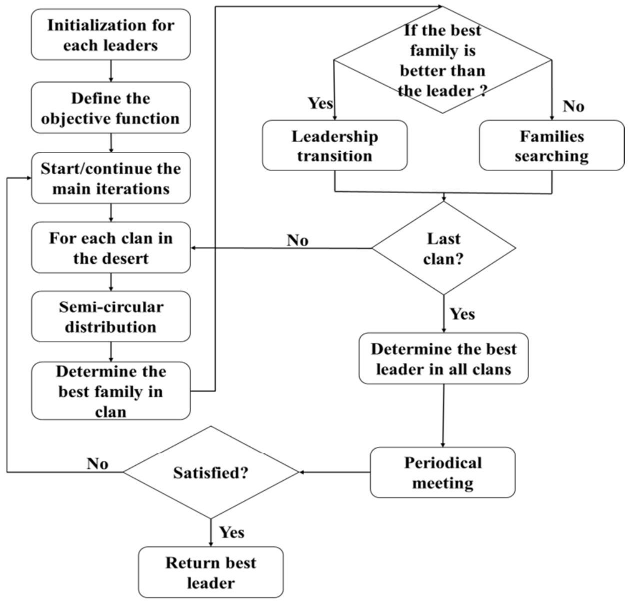
Initialization for each leaders
Define the objective function
Start/continue the main iterations
For each clan in the desert
Semi-circular distribution
Determine the best family in clan
If the best family is better than the leader ?
Yes
No
Leadership transition
Families searching
Last clan?
No
Yes
Determine the best leader in all clans
Periodical meeting
Satisfied?
No
Yes
Return best leader

Fig. 3. NPO algorithm flowchart.

Step 1: Initialization

Initialize the leaders randomly, $\sigma_i = \{\sigma_1, \sigma_2, \dots, \#c\}$ using (3):

$$\overrightarrow{\sigma_c} = (UB - LB) \times R + LB \tag{3}$$

Step 2: Semi-circular distribution (exploitation – local search)

The distribution of families, where $X_i = \{X_1, X_2, \dots, \#x\}$ around the leader σ. The distribution equations are:

$$X = \left(Rd \times \sqrt{R_1}\right) \times \cos(\theta) + X_o \tag{4}$$

$$Y = \left(Rd \times \sqrt{R_2}\right) \times \sin(\theta) + Y_o \tag{5}$$

In the scenario of finding the family position, (6) is used to find the X coordinate:

$$\overrightarrow{x_c} = \overrightarrow{\sigma_c} \times \sqrt{R} \times \cos(\theta) \tag{6}$$

Step 3: Family searching (exploration – global search)

When there is no new best local solution in NPO, global search is considered. The families start to search randomly far from the current best local solution in various directions. The random steps are based on the Lévy flight as shown in (7):

$$\overrightarrow{X_\iota^{new}} = \overrightarrow{X_\iota^{old}} + \left(a_c \times \left(\overrightarrow{\sigma_c - X_\iota^{old}}\right) \oplus \text{Lévy}\right) \tag{7}$$

The area of a clan (a_c) can calculated by:

$$a_c = \frac{\sum_{i=1}^{\Phi} \sqrt{(\overrightarrow{\sigma_c} - \overrightarrow{X_\iota^{old}})^2}}{\Phi} \tag{8}$$

The value of a_c has a great impact on the search process. When it is low, the distribution will be in a small circle. However, when the $a_c$value is high, the distribution will be in a large space (far from the current a_c). The family search uses Lévy flight for moving to another space in different directions. Lévy flight (λ_c) formula is illustrated in (9):

$$\text{Lévy} \backsim \text{u} = t^{-\lambda} \;\; (1 \le \lambda \le 3) \tag{9}$$

Step 4: Transition of Leadership (Exploitation)

The transition of the leadership is done when there is a fitness value of a new family in the clan better than the fitness value of the current leader. That family becomes the leader of the clan.

Step 5: Meeting Room Approach (MRA)

The procedure of periodical meetings is similar to the initialization except the leader distribution. The meeting occurs between the leaders (normal leaders), and the best leader (with the best solution) guides the other leaders to find the better locations with the best solutions. The variance between the leader positions is by:

$$\Delta Pos = \Psi \left(\frac{\sqrt{\sum_i^D (\sigma^E - \sigma_c^N)^2}}{\#D}\right) \tag{10}$$

For guiding the normal leaders to better positions, (11) is used to find the direction Ψ:

$$\Psi = \begin{cases} 1 & \text{if} \left(f(\sigma^E\right) \ge 0 \\ -1 & \text{otherwise} \end{cases} \tag{11}$$

After that, the normal leaders update their position by (12):

$$\overrightarrow{\sigma_c^{new}} = \overrightarrow{\sigma_c^N} + \Delta Pos(\sigma^E - \sigma_c^N) \times \frac{IT}{\#T} \tag{12}$$

The positions of all normal leaders are updated in MRA. In case a leader has a better new position than the old one, the leader will remain in its current position. The pseudo code of NPO follows.

```
ALGORITHM 1: STANDARD NPO
Input: Clans(#Clans), Families ( Φ ), Iterations(#T)
Output: best leader $\sigma^E$
Determine the NPO parameters:
Define the fitness function $f(x)$
Initialize the leaders $\sigma_c^o$ = {1, 2, 3, … , #Clans}
Find the fitness value for each leader using $f(x)$
Repeat (Iteration)
For c = 1 to #Clans
Apply semi-circular distribution using Eq. 6
Find the fitness value for each solution $x_i^c$ using $f(x)$
Set the best $x_i^c$ in the c as $\sigma_c^B$
IF $\sigma_c^B$ better than $\sigma_c^o$ then $\sigma_c^o = \sigma_c^B$
Else Explore the search space:
Find the avg. distance between all families using Eq. 8
Move family into the new position using Eq. 7
Find the fitness value for each solution $x_i^c$ using $f(x)$
Set the best $x_i^c$ in the c as $\sigma_c^B$
IF $\sigma_c^B$ better than $\sigma_c^o$ then $\sigma_c^o = \sigma_c^B$
End IF
End For
Apply MRA
Loop Until (Iteration > #T)
Return the best leader $\sigma^E$
```

E. Chaotic Maps

Chaotic maps are evolution functions that generate arbitrary patterns through exhibiting chaotic behavior. The chaotic parameters may perform in discrete-time or in continues-time. Chaotic maps are a kind of pseudo randomness, however, since they are derived from a chaotic system, most of them would be bounded, irregular, and sensitive to the initial conditions. In this work, Logistic map is used (Figure 4).

The logistic map is a classical chaotic map from the nonlinear dynamic biological population evidencing chaotic behavior [40].

$$x_{k+1} = ax_k(1 - x_k) \quad (13)$$

The initial value is x_0 = {0, 0.25, 0.5, 0.75, 1} and the map generates chaos with values within the [0,1] interval.

F. Cauchy Mutation

Cauchy mutation is a type of mutation operator used in optimization algorithms, particularly in evolutionary and genetic algorithms. It is based on the Cauchy distribution rather than the more common Gaussian (normal) distribution. This mutation operator is designed to introduce diversity into a population by allowing larger, more frequent mutations compared to Gaussian mutations, which makes it particularly useful for escaping local optima in optimization problems.

A random jumping method using the Cauchy mutation is added to the NOP to improve the ability to jump out of local optima, increasing population diversity and enhancing the capacity of the search to leave local optima. Introducing the Cauchy mutation enhances the capacity for exploitation in relation to family search and keeps the family from disintegrating into the neighborhood. The Cauchy mutation has the potential to yield different random variables that adhere to the Cauchy distribution to revise the position of the leaders and improve the search-ability of clans. The benefit of the Cauchy mutation is that it produces high-probability disruption to enhance global utilization capacity. Therefore, employing the Cauchy variation technique makes escaping from a local extreme value simpler. Equation (14) represents the cumulative distribution function of the Cauchy distribution and correlates with the probability density function of the standard Cauchy distribution function:

$$F(x) = \frac{1}{\pi}\arctan\left(\frac{x}{t}\right) + 0.5 \quad (14)$$

where $t > 0$ is a scale parameter [42]. For enhancing the distribution and diversity strategies in NPO, (15) is used:

$$\overrightarrow{X_c} = \overrightarrow{\sigma_c} \times \frac{1}{\pi}\arctan\left(\frac{x}{t}\right) + 0.5 \quad (15)$$

where $\overrightarrow{X_c}$ represents the position of a family, $\overrightarrow{\sigma_c}$ represents the position of the leader for the same swarm or clan. Equation (15) is used to enhance the diversity of the family distribution in the clan. As previously mentioned, diversity is essential to metaheuristic algorithms since it strengthens the population's ability to search for the global optimum.

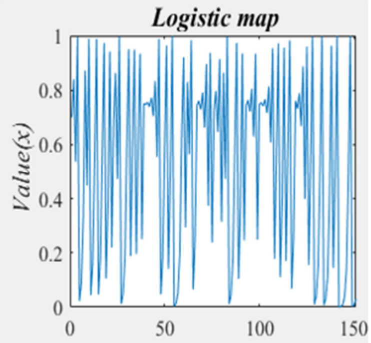

Fig. 4. Logistic map.

G. Support Vector Machine (SVM) [43]

SVM is a supervised ML model used for classification, regression, and outlier detection. SVM separates data into classes via finding the optimal hyperplane between the data, maximizing the distance between the hyperplane (margin) and the support vectors (the closet points from a class). The hyperplane is presented in (16):

$$w^T.x + b = 0 \quad (16)$$

The prime form of SVM is shown in (17):

$$minimize\ \frac{1}{2}\ \|w\|^2 \qquad (17)$$

where:

$$y_i(w^T.x_i + b) \geq 1,\ \forall_i = 1, \dots, n \qquad (18)$$

For high dimensional problems, the SVM is solved by:

$$maximize\ \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i x_j) \quad (19)$$

where:

$$\sum_{i=1}^{n} \alpha_i y_i = 0\ \ and \quad 0 \leq \alpha_i \leq C, \qquad (20)$$
$$\forall_i = 1, \dots, n$$

However, the data may not be linearly separable in the feature space. In this case, SVM performs the kernel function to handle this issue. The kernel function maps the original data into a high dimensional space to easily find the hyperplane. The common SVM kernel functions are: linear kernel, polynomial kernel, Radial Basis Function (RBF), and sigmoid kernel. Choosing the right kernel in SVM depends on several factors such as data, complexity, model type, and parameter tuning.

H. The Proposed Feature Selection Algorithm

In this paper, an improved version of NPO based on chaotic map, and SVM called CNPO, is proposed. The proposed CNOP is employed for the feature selection task. CNOP enhances the classification accuracy by selecting optimal relevant features. The inspiration of improving the standard NOP starts from initializing the leaders using a logistic map. This step creates high diversity for the leaders using (13) and enhances the performance and results of standard NPO.

In the second step of CNPO, the proposed fitness function based on SVM is presented. The proposed fitness function evaluates all the generated solutions using the following formula:

$$\min f(B_i) = a \times SVM(D) + (1-a) \times \frac{selected\,features}{all\,features} \qquad (21)$$

Minimizing (21) will lead to minimum error rate for the selected features (solutions).

In the next step of NPO, the local search (exploitation) is enhanced with Cauchy mutation. The families are distributed around the leader using (15). This step makes the distribution of the families much better around the corresponding leader than the original step in (3) and (4).

In the periodical meetings, the updating equation (12) of the normal leaders is modified by adding a random number (*CR*) to enhance exploration. Equation (22) shows the normal leader position update using CNOP:

$$\overrightarrow{\sigma_c^{new}} = \overrightarrow{\sigma_c^{N}} + \Delta Pos(\sigma^E - \sigma_c^N) \times CR \times \frac{IT}{\#T} \qquad (22)$$

The overall steps of the proposed CNOP are illustrated in the following algorithm:

```
ALGORITHM 2: THE PROPOSED CNPO
Input: Clans(#Clans), Families ( Φ ), Iterations(#T)
Output: best leader $\sigma^E$
Define the fitness function $f(x)$ via Eq. 21
Initialize the leaders $\sigma_c^o$ = {1, 2, 3, … , #Clans} using logistic map Eq. 13
Find the fitness value for each leader using Eq. 21
Repeat (Iteration)
For c = 1 to #Clans
Cauchy mutation distribution using Eq. 15
Find the fitness value for each solution $x_i^c$ using Eq. 21
Set the best $x_i^c$ in the c as $\sigma_c^B$
IF $\sigma_c^B$ better than $\sigma_c^o$ then $\sigma_c^o = \sigma_c^B$
Else Explore the search space:
Find the avg. distance between all families using Eq. 8
Move family into the new position using Eq. 7
Find the fitness value for each solution $x_i^c$ Eq. 21
Set the best $x_i^c$ in the c as $\sigma_c^B$
IF $\sigma_c^B$ better than $\sigma_c^o$ then $\sigma_c^o = \sigma_c^B$
End IF
End For
Apply MRA using Eq. 22
Loop Until (Iteration > #T)
Return the best leader $\sigma^E$
```

I. Classification

In the classification stage, SVM is used to classify the selected features from the previous stage based on feature classes (labels). The input data are separated into training and testing groups using the cross validation approach: 70% of the data were selected for training and 30% for testing. The classification result will be a class label of the input features (normal data or attack types). RBF kernel is used in SVM for classification.

J. Evaluation Measurements

There are various measurements used to evaluate the classification performance. In this paper, accuracy, precision, and recall, were considered [44]:

$$\text{Accuracy}\ \frac{TP+TN}{TP+TN+FP+FN} \times 100 \qquad (23)$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100 \qquad (24)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100 \qquad (25)$$

where TP, TN, FP, and FN represent the number of True Positives, True Negatives, False Positives, and False Negatives, respectively.

III. RESULTS

The proposed work was tested on two ID datasets, NSL-KDD, and WSN-DS, in Windows 11 environment and Python programming language. Equations (23)-(25) were used for the evaluation of the method's performance. The results are illustrated in Table III. The proposed CNOP outperformed the standard NPO in the two used datasets. CNOP achieved accuracy 99.97% and 99.99 for NSL-KDD, and WSN-DS respectively, which is better by about 6% than the NPO results. Table IV shows the comparison results of various SVM kernels and Table V shows the comparison between the SVM classifier and other ML classifiers with the proposed CNOP as shown in Table V.

TABLE III. RESULTS

Method	Dataset	Accuracy %	Precision %	Recall %
NPO	NSL-KDD	93.5	92.1	91.5
	WSN-DS	92.3	91.4	91.2
CNPO	NSL-KDD	99.97	99.97	99.97
	WSN-DS	99.99	99.99	99.99

TABLE IV. SVM KERNEL RESULTS

Classifier	Kernel	Dataset	Accuracy%
SVM	Linear	NSL-KDD	98.2
		WSN-DS	98.5
	Sigmoid	NSL-KDD	98
		WSN-DS	98.2
	Polynomial	NSL-KDD	99.2
		WSN-DS	99.5
	RBF	NSL-KDD	99.97
		WSN-DS	99.99

TABLE V. RESULTS OF DIFFERENT CLASSIFIERS

Classifier	Dataset	Accuracy %
KNN	NSL-KDD	95.2
	WSN-DS	96
ANN	NSL-KDD	97.9
	WSN-DS	98.5
SVM	NSL-KDD	99.97
	WSN-DS	99.99

TABLE VI. RESULTS OF PROPOSED APPROACH AND OTHER WORKS

Ref.	Dataset	Method	Acc.
[31]	NSL-KDD	GBA	96.96%
[30]	NSL-KDD	CBIGRU ABILSTM (PACENIDS)	96.59% 94.47% 97.67%
[7]	WSN-DS	GNB+SGD	98%
[32]	UNSW-NB15	KOMIG IDS	97.14%
[19]	KDD-CUP99	Cuckoo Algorithm	89.8%
[29]	NSL-KDD	SVM TD RF	95.2% 92.7% 94.5%
[33]	UNSW-NB15 NSL-KDD CIC-IDS2017	XGBoost + Mutual Information+ Thresholding	87.63% 80.51% 99.89%
[34]	WSN-DS	HMMs+GMMs	94.55%
[35]	WSN-DS	SMOTE-Tomek	99.92%
Proposed	NSL-KDD WSN-DS	CNOP + SVM	99.97% 99.99%

Using SVM with the proposed CNOP achieves higher accuracy results than any other classifier (Table III) and RBF kernel outperforms the other SVM kernels in both (Table IV). The proposed approach was compared with other recent works in ID using various datasets and methods. The results can be seen in Table VI.

IV. PERFORMANCE ANALYSIS

In depth discussion and analysis of the results of the proposed intrusion detection system based on CNOP is presented in this paper. CNOP is proved to be superior from the standard NOP in selecting the most relevant features leading to better detection results in both utilized datasets.

SVM with RBF kernel outperformed KNN and ANN in the classification accuracy. The SVM classifier obtained the highest classification accuracy in both datasets using RBF kernels with gamma = 0.09, and c = 1.0.

Furthermore, the proposed approach outperformed several recent intrusion detection works (Table VI). The proposed approach outperformed by 1.98% the work that used GNB and SGD on WSN-DS dataset and the works that used SVM, TD, and RF by 3%-7%. The best accuracy 99.92% of the existing works was obtained on the WSN-DS dataset using SMOTE-Tomek, but the proposed approach still get the lead in accuracy by 0.06%. Also, the proposed approach outperformed the works using various ML methods in NSL-KDD by 0.1%-4% of obtained accuracy. Furthermore, the proposed approach outperformed the works that used other IDS datasets (UNSW-NB15, CIC-IDS2017, and KDD-CUP99) in terms of accuracy.

V. CONCLUSION

This paper proposed an accurate machine learning approach for intrusion detection in networks and wireless sensor networks using chaotic maps, Cauchy mutation, NPO, and SVM classifier. The proposed approach consists of several stages, namely data collection, pre-processing, feature selection, classification, and evaluation. Two open source IDS datasets, NSL-KDD and WSN-DS, were used in the experiments. High detection results were obtained using the proposed CNPO feature selector and the SVM classifier. The best obtained results for NSL-KDD dataset were: 99.97% accuracy, 99.97% precision, and 99.97% recall. The best obtained results for the WSN-DS dataset were 99.99% accuracy, 99.99% precision, and 99.99% recall. In addition, the proposed approach was compared with recent IDS works and outperformed them.

Future studies can focus at other approaches to improve the NPO exploration and exploitation processes by enhancing the initial population in order to reach an optimal outcome.

ACKNOWLEDGMENT

We gratefully acknowledge the Lebanese University and Imam Ja'afar Al-Sadiq University for providing research facilities and resources. Their assistance was instrumental in the completion of this research project.

NOTATION LIST

Notation	Meaning
x	Original value in dataset
x_{Scaled}	Normalized value
UB	Upper Bound
LB	Lower Bound
Rand	Random value between [0,1]
$\overrightarrow{\sigma_c}$	The leader position of clan
Xo, Yo	The coordinates of origin point in circle
R1, R2	Random coordinates of a point in circle
θ	Angle value
$\overrightarrow{X_c}$	Family position
R	Random value in [0,1]
$\overrightarrow{X_\iota^{new}}$	New position of the family
$\overrightarrow{X_\iota^{old}}$	Old position of the family
ac	Clan area
Φ	Number of families in each clan
x_c^i	The normal families
λ_c	Lévy flight
σ^E	Best leader position
σ_c^N	Normal leaders position
$\#D$	Number of dimensions
ΔPos	The normalized distance
$\overrightarrow{\sigma_c^{new}}$	New position of the normal leader
IT	Current iteration
#T	Total number of iterations
x_n	Current chaotic value
x_{n+1}	Chaotic value at next iteration
W	Normal vector
b	Offsite distance
α	Lagrange multiplier
a, *CR*	Random number in [0,1]
$f(B_i)$	Fitness function
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

REFERENCES

[1] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," *International Journal of Information Security*, vol. 22, no. 1, pp. 235–247, Feb. 2023, https://doi.org/10.1007/s10207-022-00634-2.

[2] O. Abu Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, "An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning," *Expert Systems with Applications*, vol. 213, Mar. 2023, Art. no. 118745, https://doi.org/10.1016/j.eswa.2022.118745.

[3] H. Yang, J. Xu, Y. Xiao, and L. Hu, "SPE-ACGAN: A Resampling Approach for Class Imbalance Problem in Network Intrusion Detection Systems," *Electronics*, vol. 12, no. 15, Jan. 2023, Art. no. 3323, https://doi.org/10.3390/electronics12153323.

[4] A. Singh, P. K. Chouhan, and G. S. Aujla, "SecureFlow: Knowledge and data-driven ensemble for intrusion detection and dynamic rule configuration in software-defined IoT environment," *Ad Hoc Networks*, vol. 156, Apr. 2024, Art. no. 103404, https://doi.org/10.1016/j.adhoc.2024.103404.

[5] N. Jeffrey, Q. Tan, and J. R. Villar, "A hybrid methodology for anomaly detection in Cyber–Physical Systems," *Neurocomputing*, vol. 568, Feb. 2024, Art. no. 127068, https://doi.org/10.1016/j.neucom.2023.127068.

[6] J. Azimjonov and T. Kim, "Stochastic gradient descent classifier-based lightweight intrusion detection systems using the efficient feature subsets of datasets," *Expert Systems with Applications*, vol. 237, Mar. 2024, Art. no. 121493, https://doi.org/10.1016/j.eswa.2023.121493.

[7] H. M. Saleh, H. Marouane, and A. Fakhfakh, "Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning," *IEEE Access*, vol. 12, pp. 3825–3836, Jan. 2024, https://doi.org/10.1109/ACCESS.2023.3349248.

[8] E. Osa, P. E. Orukpe, and U. Iruansi, "Design and implementation of a deep neural network approach for intrusion detection systems," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 7, Mar. 2024, Art. no. 100434, https://doi.org/10.1016/j.prime.2024.100434.

[9] K. Cengiz, S. Lipsa, R. K. Dash, N. Ivković, and M. Konecki, "A Novel Intrusion Detection System Based on Artificial Neural Network and Genetic Algorithm With a New Dimensionality Reduction Technique for UAV Communication," *IEEE Access*, vol. 12, pp. 4925–4937, Jan. 2024, https://doi.org/10.1109/ACCESS.2024.3349469.

[10] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, "FlowTransformer: A transformer framework for flow-based network intrusion detection systems," *Expert Systems with Applications*, vol. 241, May 2024, Art. no. 122564, https://doi.org/10.1016/j.eswa.2023.122564.

[11] B. Mopuru and Y. Pachipala, "Advancing IoT Security: Integrative Machine Learning Models for Enhanced Intrusion Detection in Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 14840–14847, Aug. 2024, https://doi.org/10.48084/etasr.7641.

[12] H. Mamdouh Farghaly and T. Abd El-Hafeez, "A high-quality feature selection method based on frequent and correlated items for text classification," *Soft Computing*, vol. 27, no. 16, pp. 11259–11274, Aug. 2023, https://doi.org/10.1007/s00500-023-08587-x.

[13] F. Macedo, R. Valadas, E. Carrasquinha, M. R. Oliveira, and A. Pacheco, "Feature selection using Decomposed Mutual Information Maximization," *Neurocomputing*, vol. 513, pp. 215–232, Nov. 2022, https://doi.org/10.1016/j.neucom.2022.09.101.

[14] S. Rosidin, Muljono, G. Fajar Shidik, A. Zainul Fanani, F. Al Zami, and Purwanto, "Improvement with Chi Square Selection Feature using Supervised Machine Learning Approach on Covid-19 Data," in *International Seminar on Application for Technology of Information and Communication*, Semarangin, Indonesia, Sep. 2021, pp. 32–36, https://doi.org/10.1109/iSemantic52711.2021.9573196.

[15] N. O. F. Elssied, O. Ibrahim, and A. H. Osman, "A Novel Feature Selection Based on One-Way ANOVA F-Test for E-Mail Spam Classification," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 3, pp. 625–638, Jan. 2014.

[16] J. Cheng, J. Sun, K. Yao, M. Xu, and Y. Cao, "A variable selection method based on mutual information and variance inflation factor," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 268, Mar. 2022, Art. no. 120652, https://doi.org/10.1016/j.saa.2021.120652.

[17] N. Manju, B. S. Harish, and V. Prajwal, "Ensemble Feature Selection and Classification of Internet Traffic using XGBoost Classifier," *International Journal of Computer Network and Information Security*, vol. 11, no. 7, pp. 37–44, 2019, https://doi.org/10.5815/ijcnis.2019.07.06.

[18] M. K. Alsmadi *et al.*, "Intrusion Detection Using an Improved Cuckoo Search Optimization Algorithm," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 15, no. 2, pp. 73–93, Jun. 2022, https://doi.org/10.58346/JOWUA.2024.I2.006.

[19] H. Lafta, "Network Intrusion Detection Using Optimal Perception with Cuckoo Algorithm," *Wasit Journal for Pure sciences*, vol. 3, no. 1, pp. 95–105, Mar. 2024, https://doi.org/10.31185/wjps.326.

[20] M. Ragab, S. M. Alshammari, and A. S. Al-Malaise Al-Ghamdi, "Modified Metaheuristics with Weighted Majority Voting Ensemble Deep Learning Model for Intrusion Detection System," *Computer Systems Science and Engineering*, vol. 47, no. 2, pp. 2497–2512, 2023, https://doi.org/10.32604/csse.2023.041446.

[21] M. Jeyaselvi *et al.*, "A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks," *Cluster Computing*, vol. 26, no. 1, pp. 559–574, Feb. 2023, https://doi.org/10.1007/s10586-022-03607-1.

[22] T. R. Ramesh, T. Jackulin, R. A. Kumar, K. Chanthirasekaran, and M. Bharathiraja, "Machine Learning-Based Intrusion Detection: A Comparative Analysis among Datasets and Innovative Feature Reduction for Enhanced Cybersecurity," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 12s, pp. 200–206, Jan. 2024.

[23] B. Mohammed and E. K. Gbashi, "Intrusion Detection System for NSL-KDD Dataset Based on Deep Learning and Recursive Feature Elimination," *Engineering and Technology Journal*, vol. 39, no. 7, pp. 1069–1079, Jul. 2021, https://doi.org/10.30684/etj.v39i7.1695.

[24] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, "An Intrusion Detection System on The Internet of Things Using Deep Learning and Multi-objective Enhanced Gorilla Troops Optimizer," *Journal of Bionic Engineering*, vol. 21, no. 5, pp. 2658–2684, Sep. 2024, https://doi.org/10.1007/s42235-024-00575-7.

[25] M. Hasanah, R. A. Putri, M. Aidie, R. Putra, and T. Ahmad, "Analysis of Weight-Based Voting Classifier for Intrusion Detection System," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 2, pp. 190–200, 2024, https://doi.org/10.22266/ijies2024.0430.17.

[26] S. Q. Salih and A. A. Alsewari, "A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer," *Neural Computing and Applications*, vol. 32, no. 14, pp. 10359–10386, Jul. 2020, https://doi.org/10.1007/s00521-019-04575-1.

[27] S. T. Ahmed and S. M. Kadhem, "Optimizing Alzheimer's disease prediction using the nomadic people algorithm," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 2, pp. 2052–2067, Apr. 2023, https://doi.org/10.11591/ijece.v13i2.pp2052-2067.

[28] A. Q. Mohammed, K. A. Al-Anbarri, and R. M. Hannun, "Introducing newly developed Nomadic People Optimizer (NPO) algorithm to find optimal sizing of a hybrid renewable energy," vol. 928, Nov. 2020, Art. no. 022052, https://doi.org/10.1088/1757-899X/928/2/022052.

[29] B. R. Maddireddy and B. R. Maddireddy, "A Comprehensive Analysis of Machine Learning Algorithms in Intrusion Detection Systems.", *Journal of Environmental Sciences and Technology (JEST),* Vol. 3, No. 1, pp. 877-893, 2024.

[30] N. Girubagari and T. N. Ravi, "Parallel ABILSTM and CBIGRU Ensemble Network Intrusion Detection System," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 1, pp. 93–107, Feb. 2024, https://doi.org/10.22266/ijies2024.0229.10.

[31] S. S. Issa, S. Q. Salih, Y. D. Salman, and F. H. Taha, "An Efficient Hybrid Filter-Wrapper Feature Selection Approach for Network Intrusion Detection System," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 6, pp. 261–273, Dec. 2023, https://doi.org/10.22266/ijies2023.1231.22.

[32] A. S. Afolabi and O. A. Akinola, "Network Intrusion Detection Using Knapsack Optimization, Mutual Information Gain, and Machine Learning," *Journal of Electrical and Computer Engineering*, vol. 2024, no. 1, 2024, Art. no. 7302909, https://doi.org/10.1155/2024/7302909.

[33] M. A. Faizin, D. T. Kurniasari, N. Elqolby, M. A. R. Putra, and T. Ahmad, "Optimizing Feature Selection Method in Intrusion Detection System Using Thresholding," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 214–226, 2024, https://doi.org/10.22266/ijies2024.0630.18.

[34] A. R. A. Moundounga and H. Satori, "Stochastic Machine Learning Based Attacks Detection System in Wireless Sensor Networks," *Journal of Network and Systems Management*, vol. 32, no. 1, Dec. 2023, Art. no. 17, https://doi.org/10.1007/s10922-023-09794-5.

[35] Md. A. Talukder, S. Sharmin, M. A. Uddin, M. M. Islam, and S. Aryal, "MLSTL-WSN: machine learning-based intrusion detection using SMOTETomek in WSNs," *International Journal of Information Security*, vol. 23, no. 3, pp. 2139–2158, Jun. 2024, https://doi.org/10.1007/s10207-024-00833-z.

[36] R. ZHAO, "NSL-KDD." IEEE, Feb. 02, 2022, [Online]. Available: https://ieee-dataport.org/documents/nsl-kdd-0.

[37] J. Pan, Y. Zhuang, and S. Fong, "The Impact of Data Normalization on Stock Market Prediction: Using SVM and Technical Indicators," in *International Conference on Soft Computing in Data Science*, Kuala Lumpur, Malaysia, Sep. 2016, pp. 72–88, https://doi.org/10.1007/978-981-10-2777-2_7.

[38] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks," *Journal of Sensors*, vol. 2016, no. 1, 2016, Art. no. 4731953, https://doi.org/10.1155/2016/4731953.

[39] Z.-M. Gao, J. Zhao, Y.-J. Zhang, Z.-M. Gao, J. Zhao, and Y.-J. Zhang, "Review of chaotic mapping enabled nature-inspired algorithms," *Mathematical Biosciences and Engineering*, vol. 19, no. 8, pp. 8215–8258, 2022, https://doi.org/10.3934/mbe.2022383.

[40] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, Mar. 2014, https://doi.org/10.1016/j.jocs.2013.10.002.

[41] R. F. Tate, "Correlation Between a Discrete and a Continuous Variable. Point-Biserial Correlation," *The Annals of Mathematical Statistics*, vol. 25, no. 3, pp. 603–607, 1954.

[42] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, Jul. 1999, https://doi.org/10.1109/4235.771163.

[43] M. Hosseinzadeh, A. M. Rahmani, B. Vo, M. Bidaki, M. Masdari, and M. Zangakani, "Improving security using SVM-based anomaly detection: issues and challenges," *Soft Computing*, vol. 25, no. 4, pp. 3195–3223, Feb. 2021, https://doi.org/10.1007/s00500-020-05373-x.

[44] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, Jan. 2020, https://doi.org/10.1016/j.procs.2020.04.133.