Intrusion Detection Using Krill Herd Optimization Based Weighted Extreme Learning Machine

P. Kaliraj * and B. Subramani

Department of Computer Science, Shri Nehru Maha Vidyalaya (SNMV) College of Arts and Science, Coimbatore, Tamil Nadu, India
Email: pkalirajmsc@gmail.com (P.K.); drbsubramani@gmail.com (B.S)
*Corresponding author

Abstract—**With the improvement in computer network and technology, network attacks have increased drastically, as a result network intrusion becomes an important topic to work on and to find a solution to stop these network attacks. Advancement in artificial intelligence, can be utilized to find a solution for network intrusion. In this paper we are using Krill Herd Optimization (KHO) algorithm based Weighted Extreme Learning Machine (WELM) to detect the intrusion occurring in the network. Extreme Learning Machine (ELM) randomly assigns weight for neural network which is followed by the network training activity and finally the output weight is obtained. There is a need for optimization of weights used in ELM, for this purpose we are using krill herd optimization algorithm. NSL-KDD dataset is used to compare and analyze the performance of the model proposed in this paper. The experimental results show that krill herd optimization based on WELM performed better in identifying the intrusion in the network and minimize the false positive and false negative rates.**

Keywords—**Krill Herd Optimization (KHO), Weighted Extreme Learning Machine (WELM), intrusion detection in networking, false alarm reduction in networking**

I. INTRODUCTION

Let us discuss the need for intrusion detection system, there is no network or firewall that is impenetrable. In order to get around our defenses, attackers continuously develop new exploits and attack strategies. Many assaults influence other malware or social designing to acquire client accreditations that award them admittance to our organization and information. Since it enables us to identify and respond to malicious traffic, a Network Intrusion Detection System (NIDS) is essential to network security. The primary advantage of an intrusion detection system is ensuring that IT personnel are informed of potential attacks or network intrusions. A Network Intrusion Detection System (NIDS) keeps an eye on both inbound and outbound network traffic as well as data transfers between network systems. When known threats or suspicious activity are detected, the network IDS sends out alerts to IT personnel so they can investigate further and take the necessary measures to stop an attack.

Extreme Learning Machine (ELM) is a kind of training Single-hidden Layer Feed-forward Neural Networks (SLFNN) proposed by Huang *et al.* [1]. ELM randomly generates the weights of the input layer and biases of the hidden layer nodes, and determines the weights of the output layer using an analytical method. Huang *et al.* [2] have proved that ELM has the ability of consistent approximation. In fact, in the ELM algorithm, the role of the input layer to the hidden layer is a random mapping, which maps the sample points in the training set from the original space to a feature space. The dimensionality of the feature space is determined by the number of hidden layer nodes. Generally, the dimensionality of this feature space is higher than that of the original space. Compared with other SLFNNs training algorithms (such as backpropagation algorithm [3], support vector machine algorithm [4]), the advantage of ELM is that it does not need to iteratively adjust the weight parameters, and has a fast learning speed and very good generalization ability.

In recent years, ELM has become one of the research hotspots in the field of machine learning. Different researchers have proposed different ELM extension algorithms, such as Incremental ELM (I-ELM) [5] for incremental learning problems, Online Sequential ELM (OS-ELM) [6] for online sequence learning problems, and network structure Optimally Pruned ELM (OP-ELM) [7] for learning problems, etc. In 2004, Huang *et al.* extended the ELM algorithm to Radial Basis Function Neural Networks (RBFNNs) [8], and proposed the RBF-ELM algorithm. This algorithm is an algorithm for training RBF network. Its basic idea is the same as that of ELM algorithm. It randomly generates the center parameter and width parameter of radial basis function, and determines the weight of RBF network by analytical method. RBF-ELM has all the advantages of ELM, such as fast learning speed and strong generalization ability. With the rapid development of

Manuscript received May 12, 2023; revised July 17, 2023; accepted August 29, 2023; published January 25, 2024.

network technology, the network structure tends to be complex, and the risk of network intrusion is also increasing. How to identify various network intrusions has become a problem of great concern. Intrusion Detection (ID) technology as a new type of security mechanism that can dynamically monitor, prevent and defend against intrusions has gradually developed into a key technology to ensure the security of network systems. However, the continuous increase, increase of network scale, network speed and intrusion types make intrusion detection technology more and more difficult [9]. Traditional neural networks such as Backpropagation (BP) neural networks have been applied to network intrusion detection in many literatures, and have achieved certain results [10, 11], but it needs multiple iterations to determine the network output weights, which seriously affects the learning ability of the network. As a new type of neural network, the emergence of Extreme Learning Machine (ELM) [12] has attracted extensive attention of researchers.

Raman *et al.* [13] have shown that in swarm intelligence search, the convergence performance of the algorithm will be affected by the initial population. The larger the population and more uniform the distribution, the more the algorithm can converge to the optimal solution in a shorter time; conversely, it will affect the convergence performance of the algorithm. The fitness function is the criterion for judging the ability of individuals to adapt to the environment. Therefore, the selection of the fitness function directly affects the convergence accuracy of the algorithm and whether the optimal solution can be found. Mazini *et al.* [14] takes the intrusion detection accuracy, false positive rate, and characteristic number. The proportional weight of the swarm intelligence algorithm is used as the fitness function, which not only increases the amount of calculation but also leads to low convergence accuracy due to improper counting;

Zong *et al.* [15] regards the segmental function of the function value of the swarm intelligence algorithm as the fitness function, and needs to determine the fitness through the function value expressions, which increase the computational complexity. The krill swarm algorithm is a new type of bionic swarm intelligence algorithm proposed by Xiao [16], which is a simplification of krill swarm behavior and swarm intelligence interaction. Basic behaviors: induced movement; foraging movement and perturbation behavior that solves optimization problems through this swarm intelligence. The krill swarm algorithm has the advantages of simplicity, easy implementation, good convergence and strong robustness. In recent years, researchers have obtained some research results and experience in improving the krill swarm optimization algorithm, which provides a reliable basis for the development of krill swarm optimization algorithm and guides a new direction for practical engineering applications. Liu *et al.* [17] proposes a chaotic krill swarm algorithm, using singer map chaotic mapping to generate inertia weights and adding elite strategy at the same time, replacing the worst individual with the globally optimal individual, which improves the reliability of the global optimal and the quality of the solution. Liu *et al.* [18] proposed an improved krill swarm algorithm based on natural selection and random disturbance. The algorithm uses a nonlinear decreasing strategy to calculate the induction weight and foraging weight, and adds a random disturbance factor when generating a new generation of krill populations, thereby effectively improving the quality of individuals in the krill population, and improving the global search and local exploration capabilities.

Wang *et al.* [19] combines the cuckoo algorithm with the krill swarm algorithm and adopts a greedy selection scheme. After each iteration, the algorithm discards some of the worst krill and replaces them with randomly generated new krill, thus maintaining the diversity of the population and improving the optimization ability of the algorithm. Although the research of krill swarm algorithm has made many important progress, but krill swarm algorithm has some disadvantages such as easy to fall into slow convergence speed, local optimum, etc. Therefore, the improvement research on krill swarm algorithm has become an important topic. In this paper, the orthogonal diagonalization strategy is used to deal with the location update of krill individuals, which can better improve the shortcomings of the krill swarm algorithm. Because of its simple programming and easy implementation, it has been applied in the fields of power system, neural network training, optimal scheduling, and cluster analysis, etc.

This paper is divided into five sections: Section I gives the introduction, Section II deals with krill algorithm, Section III is analyzing the Weighted Extreme Learning Machine, Section IV is discussing the WELM based KHO, Section V imparts algorithm simulation and result analysis, Section VI draws the conclusion finally reference taken for this paper is discussed.

II. Krill Herd Optimization Algorithm

In the krill algorithm, the position change of krill individuals mainly depends on the following three aspects: 1) the induced movement between krill individuals; 2) the foraging movement of krill individuals; 3) the random diffusion movement of krill individuals. The formula is as follows:

$$\Delta X_i = N_i + F_i + D_i \tag{1}$$

In Eq. (1): F_i is the position change caused by the foraging movement of the krill individual; ΔX_i is the position change of the i-th krill individual which is shown in Eq. (1); N_i is the position change caused by the induced movement of the krill individual; D is the random diffusion movement of the individual krill caused by the change in position.

A. Induced Movement among Krill Individuals

The behavior of krill individuals affected by other individuals can be expressed for:

$$N_i^{\text{new}} = N^{max}\boldsymbol{\alpha}_i + w_n N_i^{\text{old}} \tag{2}$$

$$\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_i^{\text{local}} + \boldsymbol{\alpha}_i^{\text{target}} \tag{3}$$

N^{max} is the maximum induced velocity; $\boldsymbol{\alpha}_i$ is the individual swimming direction vector as given in Eq. (3); $\boldsymbol{\alpha}_i^{\text{local}}$ is the vector sum of the motion influence of the krill adjacent individuals; $\boldsymbol{\alpha}_i^{\text{target}}$ is the direction vector provided by the optimal individual; N_i^{old} is the individual position generated last time change; w_n is the induced weight and the value of range is [0,1]. Eq. (2) represents the new position which is represented as N_i^{new}.

In a krill population, the influence between adjacent individual is defined as follows:

$$\boldsymbol{\alpha}_i^{\text{local}} = \sum_{j=1}^{NN} \widehat{K}_{ij}\widehat{\boldsymbol{X}}_{ij} \tag{4}$$

$$\widehat{K}_{ij} = \frac{K_i - K_j}{K^{\text{worst}} - K^{\text{hest}}} \tag{5}$$

$$\widehat{X}_{ij} = \frac{X_j - X_i}{\|X_j - X_i\| + \varepsilon} \tag{6}$$

In the formula: K^{worst} is the worst fitness value of the krill group so far; K^{hest} is the best fitness so far; K_i is the witness value of i-th krill; K_j is the witness value of j-th krill individual; X_i, X_j is the position of the krill individual; $\widehat{\boldsymbol{X}}_{ij}$ is the distance of krill individuals which is given in Eq. (6); in order to avoid possible abnormalities, a small integer ε is added to the denominator; NN is the number of adjacent individual, $\boldsymbol{\alpha}_i^{\text{target}}$ can be expressed as:

$$\boldsymbol{\alpha}_i^{\text{target}} \& = C^{\text{best}}\widehat{K}_{i,\,\text{best}}\widehat{\boldsymbol{X}}_{i,\,\text{best}} \tag{7}$$

$$C^{\text{best}} \& = 2\left(\text{rand} + \frac{I}{I_{max}}\right) \tag{8}$$

In the formula: rand is a random number between 0 and 1; I is the current number of iterations; I_{max} is the maximum number of iterations.

B. Foraging Movement of Individual Krill Position

Change caused by the foraging behavior of individual krill can be expressed as

$$\boldsymbol{F}_i = V_f\left(\boldsymbol{\beta}_i^{\text{food}} + \boldsymbol{\beta}_i^{\text{best}}\right) + w_f\boldsymbol{F}_i^{\text{old}} \tag{9}$$

In the formula V_f is the forging speed; w_f is the foraging weight, the value range is [0,1]; $\boldsymbol{F}_i^{\text{old}}$ is the previous krill individual foraging movement vector; $\boldsymbol{\beta}_i^{\text{food}}$ is the impact factor of the food position on the i-th krill, which can be expressed as:

$$\boldsymbol{\beta}_i^{\text{food}} = 2\left(1 - \frac{I}{I_{max}}\right)\widehat{K}_{i,\text{food}}\widehat{\boldsymbol{X}}_{i,\text{food}} \tag{10}$$

$$X_{\text{food}} = \frac{\sum_{i=1}^{N}\frac{1}{K_i}X_i}{\sum_{i=1}^{N}\frac{1}{K_i}} \tag{11}$$

$\widehat{K}_{i,\text{food}}\widehat{\boldsymbol{X}}_{i,\text{food}}$ can be expanded according to Eqs. (5) and (6); $\boldsymbol{\beta}_i^{\text{food}}$ is the influence factor of the optimal fitness position of the i-th krill as shown in Eq. (10), which can be expressed as

$$\boldsymbol{\beta}_i^{\text{best}} = \widehat{K}_{i,i\,\text{best}}\widehat{X}_{i,\,\text{ibest}} \tag{12}$$

C. Krill Individual Random Diffusion

The physical random diffusion process of individual krill D_i given in Eq. (13) can be expressed as

$$D_i = D^{max}\left(1 - \frac{I}{I_{max}}\right)\boldsymbol{\delta} \tag{13}$$

where D^{max} is the maximum diffusion velocity; $\boldsymbol{\delta}$ is the random vector direction $\boldsymbol{\delta}$ ranges from [–1,1].

III. Weighted Extreme Learning Machine

Based on ELM, this paper establishes a weighted extreme learning machine classification model. The network structure of the ELM model is shown in Fig. 1. In Fig. 1, suppose given N training samples $\{x_i, t_i\}_{i=1}^{N}$, $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$, $t_i = [t_{i1}, t_{i2}, \ldots, t_{in}]^T \in R^m$, where n is the number of features of the sample, m is the number of categories of the sample. A feedforward neural network output model with L hidden layer nodes can be expressed as follows:

$$\sum_{h=1}^{L}\beta_h G(a_h, b_h, x_i) = t_{i,i=1,2,\ldots,N} \tag{14}$$

where: G is the activation function of the hidden layer neurons; β_h is the output weight of the h^{th} hidden layer neuron; o_i for the first actual output value of i training samples; a_h, b_h respectively h individual Input weights and biases of hidden layer neurons; x is the input sample, t_i for the first Expected output for i training of samples.

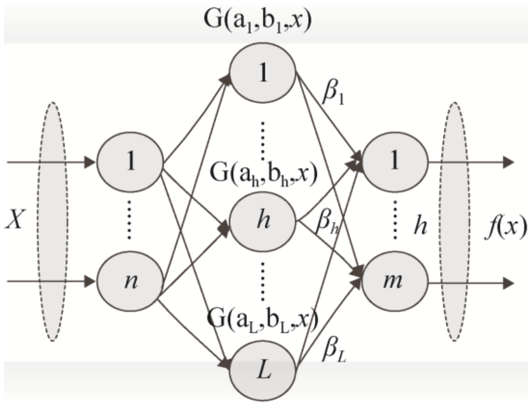

Fig. 1. Basic structure diagram of single hidden layer feedforward neural network.

According to Ref. [15], for a quantity of N training samples, $\{x_i, t_i\}_{i=1}^{N}, x_i \in R^n$ There exists a (a_h, b_h) and β_h with $\sum_{i=1}^{L}\|o_i - t_i\| = 0$, so that the single-hidden layer feedforward neural network model can approximate the training set with zero error $\{x_i, t_i\}_{i=1}^{N}, x_i \in R^n$ which is,

$$\sum_{h=1}^{L} \beta_h G(a_h, b_h, x_i) = t_{i,i=1,2,\ldots,N}$$

The equation can be further simplified as*: Hβ=T.*
where H is the output matrix of the hidden layer; β is the output weight of the hidden layer matrix; T is the expected output matrix corresponding to the training samples.

During the training process of ELM, when initializing the network parameters, the input weights a_h of the hidden layer and the bias b_h of the hidden layer are randomly generated, and remain unchanged during the whole training and testing process. Since the input training samples, the input weights and biases of the hidden layer, and the expected output are all known, the entire training process is to obtain the hidden layer output weight matrix β in the ELM model, so as to obtain a complete classification model.

From the Moore-Penrose generalized inverse matrix H^+ of the hidden layer output matrix H, we can get,

$$\hat{\beta} = H^+ T \tag{15}$$

In the formula: There are many ways to calculate H^+. In ELM, the orthogonal projection method (KKT) is usually used to solve H^+. When $H^T H$ is a non-singular matrix, $H^+=(H^T H)^{-1} HT$; when HH^T is a non-singular matrix, $H^+=H^T(HH^T)H^{-1}$.

In order to solve Eq. (2), a sufficiently small regular term 1/C is added to the diagonal of HH or HH^T, so that the classification model has better stability and generalization performance. The output weight of the hidden layer can be expressed as,

$$\hat{\beta} = \begin{cases} H^T (I/C + HH^T)^{-1} T, \ N < 1 \\ (I/C + H^T H)^{-1} H^T T, \ N \geq 1 \end{cases} \tag{16}$$

The output function of ELM can be expressed as,

$$f(x) = h(x)\hat{\beta} = \begin{cases} h(x) H^T \left(\frac{I}{C} + HH^T\right)^{-1} T, \ N < 1 \\ h(x) \left(\frac{I}{C} + H^T H\right)^{-1} H^T T, \ N \geq 1 \end{cases} \tag{17}$$

In the classification problem, not all classified sample data are evenly distributed. In order to solve the classification problem of unbalanced samples, Zong *et al.* [15] proposed a Weighted Extreme Learning Machine (WELM) algorithm based on ELM. Xiao [16] proposes to assign weights to each sample according to the weighting scheme:

Weighting scheme one $W1$ represented in Eq. (18): Automatic weighting scheme:

$$W1 = \frac{1}{Count\ (t_i)} \tag{18}$$

where $Count(t_i)$ is the number of samples of class t_i in the training sample.

Weighting scheme two $W2$ as expressed in Eq. (19): push the ratio of the minority class to the majority class to the direction of 0.618:1 (the golden ratio). This plan exchanges for the classification accuracy of the minority class and it sacrifices the classification accuracy of the majority class.

$$W2 = \begin{cases} \frac{0.618}{Count(t_i)}, t_i \text{ belong to the majority class} \\ \frac{1}{Count\ (t_i)}, t_i \text{ belong to the minority class} \end{cases} \tag{19}$$

The output weights of the WELM hidden layer can be expressed as,

$$\hat{\beta} = H^+ T = \begin{cases} H^T (I/C + WHH^T)^{-1} WT, \ N < 1 \\ (I/C + H^T WH)^{-1} H^T WT, \ N \geq 1 \end{cases} \tag{20}$$

In the formula: the weighting matrix is a diagonal matrix of *NXN*; the *N* main diagonal elements correspond to *N* samples, and for different sample categories different weights are assigned, and the weighted weights of the same category are equal.

When the hidden layer feature map $h(x)$ is unknown, the kernel matrix is defined as,

$$\Omega_{ELM} = HH^T : \Omega_{ELMi,j} = h(x_i) h(x_j) = K(x_i, x_j) \tag{21}$$

In the formula: Ω_{ELM} is the kernel matrix, and the kernel function K needs to satisfy the Mercer condition. Common kernel functions include Gaussian kernel function, radial basis kernel function, polynomial kernel function and linear kernel function. By Eq. (21), the output function Eq. (17) can be expressed as,

$$f(x) = h(x)\hat{\beta} = h(x) H^T \left(\frac{I}{C} + WHH^T\right)^{-1} WT = \begin{bmatrix} K(x, x_1) \\ \cdot \\ \cdot \\ \cdot \\ K(x, x_n) \end{bmatrix} (I/C + W\Omega_{ELM})^{-1} WT \tag{22}$$

Therefore, the training process of the classification model based on the weighted extreme learning machine is:

(1) Randomly set the input weights ah and bias b_h of the hidden layer, where $h = 1, 2, \ldots, L$;

(2) According to the weighting scheme, assign weights to each sample, and calculate the weighting matrix W;

(3) Calculate the kernel matrix Ω_{ELM} according to the selected kernel function;

(4) Calculate the output using Eq. (22).

IV. Weighted Extreme Learning Machine Algorithm Based on Krill Herd Optimization

This paper uses feed forward neural network WELM, its positives are reduced training time, better generalization performance and the ability to optimize globally by KHO. The Krill Herd Optimization (KHO) algorithm optimizes the randomly determined hidden layer input weight supports in WELM network. Processing rectifies the data imbalance in network intrusion detection, it also enhances the recall rate of minority attacks in network attack and also restricts WELM from falling into local optimal solutions. The algorithm flow of this paper is shown in Fig. 2.

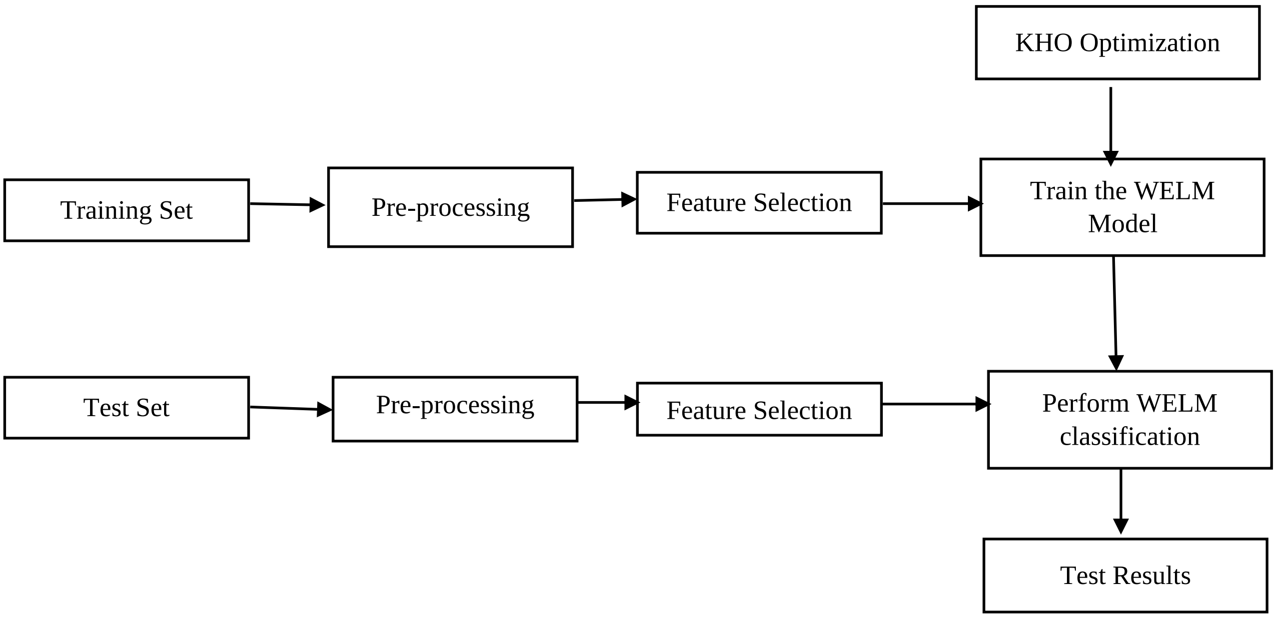

Fig. 2. Intrusion detection model using weighted ELM classification.

The algorithm flow follows below:

(1) The test set and training set are selected from the NSL-KDD dataset, and the data is preprocessed in the dataset;

(2) Inter-relation between each feature in the data set is analyzed and features with greater correlation to minimize the dimension of the data set;

(3) Start the network parameter of WELM and KHO: starting position [X axis, Y axis] of the krill herd is randomly determined, herd size N is set, the maximum number of iterations max and the step size control factor a, the hidden layer of WWLM and the input neurons is determined. The number of output neurons determine the weighting scheme, and start the biases and input weight of the WELM hidden layer.

(4) Training set is fed into WELM, and the taste concentration of each KRILL is calculated;

(5) Training set is fed into WELM, and the taste position of each KRILL is calculated;

(6) Update the direction and position of the krill, iterative optimization stage is entered;

(7) If the iterations count is more than the maximum value, position of the krill individual is saved with the absolute taste concentration (i.e.) bias and input weight of globally optimal hidden layer; or else, increase one in the number of iterations and return to step (2);

(8) Substitute the biases and absolute hidden layer input weight into WELM, and proceed with the experiments of the test set.

The advantages of KHO are as follows:

Meta-heuristics are well-suited for real problems with a large number of local solutions because of the higher local optima avoidance capability of KHO. Meta-heuristics also have the advantages of simplicity, adaptability, and derivation-free structure. General genetic operators rather than stochastic selection are used by the best krill, the Stud, to provide its optimal information to all of the other members of the population. Notwithstanding the previously mentioned benefits higher achievement rates can be accomplished when contrasted with other advancement calculations.

V. Algorithm Simulation and Result Analysis

Algorithm simulation and result analysis is done in four stages preprocessing and data set selection, performance measures, data dimensionality reduction processing and finally simulation and result analysis.

A. Pre-Processing and Data Set Selection

In this paper, for experimental data set the NSL-KDD dataset [9] is selected, KDDTest+ and KDDTrain+ in NSL-KDD data package are used as test set and training set of the experiment. Every dataset has forty-two dimensional data, of which first forty-one dimensions are dataset feature and the last dimension is dataset label bits. The tag bits include normal data and 39 types of attacks, of which the 39 types of attacks belong to four types of attacks: U2R, R2L, DOS and probe. Among them, there are twenty-one types of intrusion attack in training set, and test set consist of eighteen types of intrusion attacks. These intrusion attacks are likely to be seen in test set, that can be utilized to analyze the detection ability of the intrusion detection algorithm against unknown attacks. Table I shows the dispensation of each label class in test set and training set.

TABLE I. NUMBER OF FEATURES IN EACH SUBCLASS IN TRAINING AND TEST SET

Class	Normal	DOS	U2R	R2L	Probe	Total
Training Set	67343	45927	52	995	11656	125973
Test Set	9711	7458	200	2754	2421	22544

Before training the model, the data in the dataset needs to be pre-processed:

Convert the character type feature into the forty-one-dimensional feature of the numerical NSL-KDD dataset, the 2nd dimensional feature protocol type (Protocol Type), the 3rd dimensional feature network service (Service) and 4th dimensional feature connection The state (Flag) is a character type feature and needs to be converted into a numerical type. Denote TCP as 1, UDP as 2, and TCMP as 3 in the second dimension feature. The 67 service types in the third dimension feature are respectively recorded as 1 to 67 in the alphabetical order by their names. In the fourth dimension feature, the 11 kinds of Flag states are recorded as 1–11, respectively. In the 42nd dimension label bits, there are 5 types of labels: DOS, Normal, Probe, U2R and R2L, which are recorded as 0–4, respectively.

The numerical data set in the previous step is worked according to Eqs. (23) and (24), and the measurement units between various eigenvalues are unified to reduce the detection results caused by the difference of measurement units. influence.

$$\text{Normalized formula: } x_1 = \frac{x-\bar{x}}{\sigma} \tag{23}$$

In the formula: x is the eigenvalue; $\bar{x}$ is the mean value of the eigenvalue; σ is the standard deviation of the eigenvalue; x_1 is the standardization result of the dimension of each data sample.

The normalization formula is:

$$\text{Normalized formula: } x_2 = \frac{x_1 - x_{1min}}{x_{1max} - x_{1min}} \tag{24}$$

In the formula: $x_{1\max}$ is the highest value of all samples of this dimension feature processed by Eq. (23); $x_{1\min}$ is the lowest value of all samples of this dimension feature processed by Eq. (23); x_2 is the value of each data sample. The result after dimension feature normalization.

B. Performance Measures

In this paper, the three performance analysis indicators of false alarm rate, recall rate and precision rate are used to analyze the advantages and disadvantages of the algorithm in this paper. Accuracy and False Positive Rate (FPR) is expressed in Eqs. (25) and (26).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{25}$$

$$False\ Positive\ Rate(FPR) = \frac{FP}{FP+TN} \tag{26}$$

$$Recall = \frac{TP}{TP+FN} \tag{27}$$

where TN represents True Neutral, TP represents True Positive, FN represents False Negative and FP represents False Positive, respectively.

C. Data Dimensionality Reduction Processing

The pre-processed forty-one dimensional data features are evaluated for correlation by the method of Pearson correlation coefficient, and the weighted ranking diagram of every feature is obtained, in which according to the order of the features in the NSL-KDD data set, the ranking order is arranged as shown in Fig. 3.

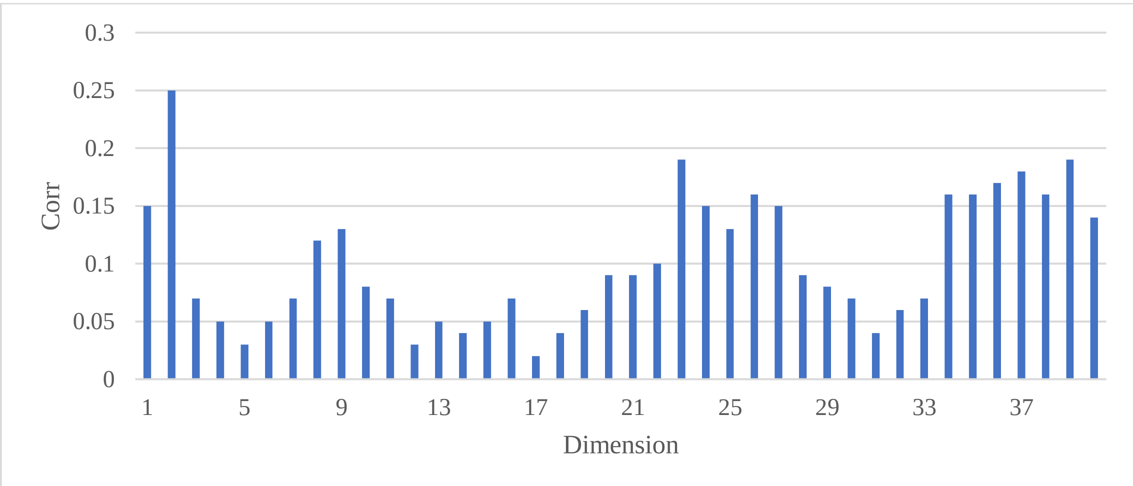

Fig. 3. Sorted graph of feature weight.

In the experiment, the features with correlation coefficient (corr) > 0.05 were taken as the test set and training set after dimension reduction. Therefore, the dimensionality-reduced test set and training set contain eighteen dimensional features.

D. Simulation and Result Analysis

The simulation environment of this paper is the inter i7 processor 3.2 GHz, 8 GB memory, which is realized by compiling a simulation program using MATLAB.

1) Comparison experiment between Weighted ELM and other machine learning algorithms

The number of Weighted ELM input layer nodes is set to eighteen, the number of output layer nodes is 5, the number of hidden layer nodes is 250 Case 2, Sigmoid kernel function is selected for the kernel function. The detection results of Random Forest, (RF), K-Nearest Neighbor (KNN) algorithm, BP neural network and Extreme Learning Machine (ELM) algorithm are compared with the algorithm in this paper. The comparison test results are shown in Table II.

It can be seen from Table II that the weighted extreme learning machine algorithm is similar to other machine learning algorithms in the detection effect of Dos, Normal and Probe and in the detection effect of U2R and R2L, the recall rate of WELM is higher than other machine learning algorithms. But in contrast, the recall rates of these two types are still relatively low, because the two major attacks, U2R and R2L, appear lower in actual conditions, so the sample size is very small in training set, and the data set is unbalanced phenomenon.

2) Relative experiment of KHO-WELM algorithm and WELM

The WELM algorithm optimized by KHO and FOA respectively is compared to the WELM algorithm without optimization. The herd size of the krill is set to 50, the number of iterations is 300, and the direction and distance of the krill are set to [–20, 20], the step size control factor α is 0.95. The number of nodes in the input layer of WELM is eighteen, the number of nodes in the hidden layer is 250, and the number of nodes in the output layer is 5. The weighting scheme selects scheme 2, and the kernel function selects the Sigmoid kernel function. The experimental results are shown in Table II.

TABLE II. PREDICTION RESULTS OF VARIOUS TECHNIQUES

Detection Model	Recall (%)					Accuracy (%)	False Alarm Rate (%)	Detection Time (sec.)
	Normal	DOS	U2R	R2L	Probe			
KNN	97	76	34	11	63	81	34	3.1
BPN	95	74	36	24	62	81	32	3.1
SVM	97	73	33	45	66	82	32	3.0
ELM	96	77	31	46	61	83	27	2.4
WELM	97	78	43	44	67	85	12	2.2
FOAELM	97	77	44	46	67	87	7	1.7
KHOELM	97	78	44	47	66	87	2	1.5

It can be seen from Table II that the WELM algorithm optimized by KHO has undergone global optimization, as a result recall rate of four types of attacks are improved. Compared with FOA-WELM, KHO-WELM has better classification results, especially the recall rate of U2R attack is increased by 6% compared with WELM, the test set classification accuracy rate is increased by 3%, and the false alarm rate is reduced by 4.5%. Under the same experimental environment, the number of iterations is 300, the population size of the krill is set to 50, the direction and distance of the fruit fly are set to [–20, 20], and the step size control factor α is set to 0.95. The training time of FOA-WELM is 1.7 s, and the training time of KHO-WELM algorithm is 1.5 s. This is because the adaptive capacity of the newly added optimization algorithm increases the time complexity, so the training time increases.

From Fig. 4, we can see that KHOELM performed better than other compared models.

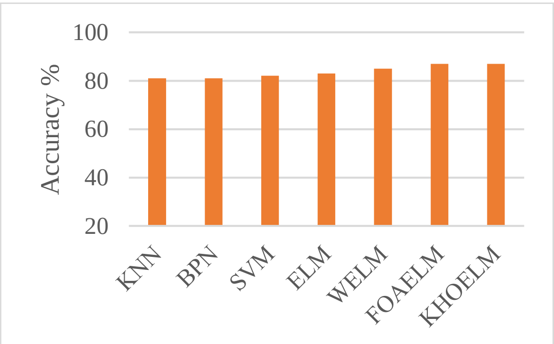

Fig. 4. Accuracy comparison of proposed model with other models.

From Fig. 5, it can be seen that proposed model outperformed other compared models when it comes to false alarm rate.

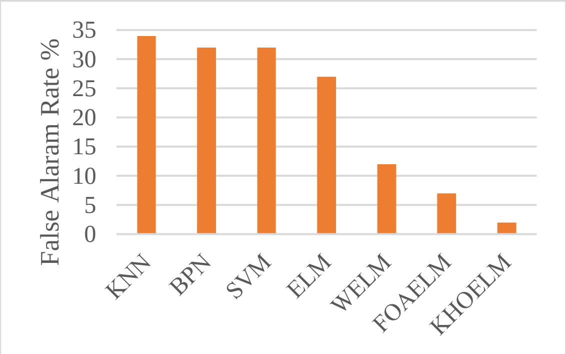

Fig. 5. False alarm rate comparison of proposed model with other models.

Fig. 6. Detection time comparison of proposed algorithm with other models.

Fig. 6 shows that KHOELM has better detection time when compared to other existing models.

VI. Conclusion

For the conclusion, we used krill herd optimization algorithm to improve the performance of WELM. From the comparison it is clear that the model proposed in this paper performed better than other traditional models when it comes to intrusion detection. Recall, False alarm rate and Detection time are decreased thereby increasing the accuracy of intrusion detection. Though some of the traditional methods performed better in some of the attacks the overall performance is significant in KHO based WELM, the data imbalance is also reduced by using this method. And so, it is efficient to use KHO based WELM to detect the intrusion in network.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

P. Kaliraj contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. P. Kaliraj and B. Subramani discussed the results and commented on the manuscript. All authors had approved the final version.

References

[1] G. B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2010.

[2] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.

[3] X. Zhang, "On statistical learning theory and support vector machines," *Acta. Automatica Sinica*., vol. 26, no. 1, pp. 32–42, 2000.

[4] X. Chen and M. Liao, "Dimensionality reduction of extreme learning machine based on sparsity and neighbor preservation," *Acta. Automatica Sinica.*, vol. 45, no. 2, pp. 325–333, 2019.

[5] L. Wei, "Improvement and application of online sequential extreme learning machine," Master's thesis, Southwest University, 2020.

[6] Y. Chen, "Research on dynamic adjustment method of hidden layer structure based on convex optimal growth limit learning machine," Master's thesis, Northeastern University, 2015.

[7] X. Wang, H. Xie, J. Wang, W. Chen, Z. Cai, and Z. Liu, "Dam deformation prediction based on Bootstrap and ICS-MKELM algorithm," *Journal of Hydroelectric Power*, vol. 39, no. 3, pp. 106–120, 2020.

[8] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. 2004 IEEE International Joint Conference on Neural networks*, IEEE, 2004, vol. 2, pp. 985–990.

[9] H. W. Ding and L. Wan, "Research on intrusion detection based on KPCA-BP neural network," in *Proc. the 18th IEEE International Conference on Communication Technology (ICCT)*, IEEE, 2018, pp. 911–915.

[10] Y. Yang, L. Li, H. Peng, J. Yuan, Y. Chen, and H. Zhang, "Swarm intelligence algorithm and its application exploration in information security," *Journal of Cyber Security Information Security Journal*, vol. 1, no. 1, 2016.

[11] H. Fu, "Semi-supervised classification based on extreme learning machine," Doctoral dissertation, Xidian University, 2013.

[12] Y. Liu and L. Shao, "Improved whale group optimization algorithm and its application," *Journal of Liaoning Technical University: Natural Science Edition*, no. 2, pp. 422–429, 2018.

[13] M. G. Raman, N. Somu, K. Kirthivasan, R. Liscano, and V. S. Sriram, "An efficient intrusion detection system based on hypergraph-genetic algorithm for parameter optimization and feature selection in support vector machine," *Knowledge-Based Systems*, vol. 134, pp. 1–12, 2017.

[14] M. Mazini, B. Shirazi, and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," *Journal of King Saud University- Computer and Information Sciences*, vol. 31, no. 4, pp. 541–553, 2019.

[15] W. Zong, G. B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[16] G. Xiao, "Chaotic krill swarm algorithm with dynamic adjustment of inertia weight," *Journal of Lanzhou University of Arts and Science: Natural Science Edition*, vol. 31, no. 6, pp. 59–63, 2017.

[17] P. Liu, Y. Gao, and W. Guo, "Improved krill swarm algorithm based on natural selection and random disturbance," *Small and Micro Computer Systems*, vol. 38, no. 8, pp. 1845–1849, 2017.

[18] J. Liu, L. Li, G. Chen, and Y. Liu, "High precision IGBT health evaluation method: Extreme learning machine optimized by improved krill herd algorithm," *IEEE Transactions on Device and Materials Reliability*, vol. 23, no. 1, pp. 37–50, 2022.

[19] Q. Wang, C. Ding, and X. Wang, "A hybrid data clustering algorithm based on improved KH and KHM clustering," *Control and Decision Making*, vol. 35, no. 10, pp. 2449–2458, 2020.