WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Improving Accuracy in Attack Detection in Computer Network Using Binary Grasshopper Algorithm and Transfer Functions

Mohammad Dawood Raqib[1], Hamed Quraishi[2]

[1,2]Lecturer of Computer Science Faculty, Network Engineering Department, Jami University

[1] Raqib1365@gmail.com – ORCID No: 0009-0009-0628-081X

[2] hamed.cs2010@gmail.com

DOI : https://doi.org/10.55248/gengpi.6.0125.0402

ABSTRACT

To address the challenges posed by attacks on computer networks and systems, various methods have been developed, notably intrusion detection systems (IDS). The primary objective of an IDS is to prevent unauthorized access, misuse, and damage to computer systems and networks caused by both internal users and external attackers. Numerous IDS have been established utilizing machine learning techniques, including algorithms like Support Vector Machines, decision trees, neural networks, and various other approaches. These techniques alone have weaknesses, such as computational complexity and long implementation time, which have caused a lack of optimal accuracy. In fact, it can be said that combining appropriate techniques is very necessary and important to increase the efficiency and diagnosis of the nature of attacks. Therefore, in this study, a combined method (simultaneous optimal selection of transfer functions and feature selection) has been presented using a binary grasshopper algorithm. The simultaneous optimal selection of these parameters can lead to an increase in accuracy, resulting in a significant improvement compared with the situation where we choose only optimal feature selection or optimal parameters. In addition, the Binary Grasshopper Algorithm (BGOA) is used to enhance the Support Vector Machine (SVM) parameters. The dataset utilized in this study is NSL-KDD, including 41 features and 5 different classes to clarify the behavior of packets in the network. According to the results achieved in this research, it is observed that the enhancement of accuracy in the proposed method is 5.26% higher than that reported in the referenced article.

Key words: Intrusion Detection System (IDS), Machine Learning Techniques, Binary Grasshopper Algorithm, Support Vector Machine (SVM), Transfer Functions.

INTRODUCTION

Due to increased cyber-attacks, network intrusion detection systems (IDS) have become an important research field. These systems protect computer networks and systems from unauthorized access and malicious activities. (Mohammed Amin Ferrag, 2020).

Based on the existing literature, IDS can be categorized into two main types: signature-based detection, which identifies network behavior through predefined signatures or patterns, and anomaly-based detection, which focuses on detecting anomalous behavior that deviates from established standards (Mohammed A. Ambussaidi, 2016). Considering the huge amount of data generated on computer networks and the complexity of modern attacks, signature-based methods have proven insufficient. Therefore, researchers have increasingly turned to anomaly-based intrusion detection techniques, which can be further categorized into data mining algorithms and artificial intelligence approaches. There has been a lot of research in the field of Intrusion Detection Systems (IDS). The main goal of much of the research in this field has been to improve the accuracy of IDS while ensuring the accuracy of detection. IDS typically provides services such as monitoring and analysis of network activity, management of system configurations and vulnerabilities, and evaluation and detection of anomalous activity. In a study published in

(Hariharan Rajadurai, 2020), a combined deep learning and PCA model (DL-PCA) was used to identify and classify attacks. Deep learning is a core component of statistical solutions that helps identify important variables in a dataset. In addition, the reduced dataset has its own features that serve as input to the aforementioned deep learning model. The dataset used in this study is NSL-KDD, which contains 41 features and 30,319 records.

researchers (F. (2019) used neural network algorithms and deep learning to detect intruders in vehicles. To improve efficiency and accuracy, approaches such as gradient descent with momentum (GDM) and gradient descent with momentum and adaptive gain (GDM/AG) have been implemented. In our experiments, we found that using the GDM/AG algorithm achieves faster convergence in vehicle anomaly detection compared to the GDM algorithm. Moreover, data anomalies can be detected within milliseconds. Moreover, the model we present can self-adapt to detect unknown attacks.

The accuracy rate for unknown attacks is between 97% and 98%. Research (Binghao Yan, 2018) investigated and compared deep learning algorithms and datasets used in intrusion detection systems (IDS). In this context, deep neural networks (DNN), restricted Boltzmann machines (RBM), recurrent

neural networks (RNN), generative adversarial networks (GAN), deep Boltzmann machines (DBM), autoencoders, and deep belief networks were compared on three main performance metrics: accuracy, false alarm rate, and detection rate using two new datasets (CSE-CIC-IDS2018 dataset and Bot-IoT dataset).

Research (Sunita S., A Hybrid Approach for Intrusion Detection Using ANN and FCM, 2016) aims to reduce unnecessary and irrelevant features using the FMIFS algorithm while controlling both linear and nonlinear characteristics. In this study, we propose the LSSVM algorithm for classification. Three datasets, KDD Cup 99, NSL-KDD, and Kyoto 2006, are used to evaluate the performance of LSSVM-IDS.

Based on the obtained results, the proposed feature reduction algorithm reduces the computational complexity of the classification algorithm while achieving higher classification accuracy, higher detection rate and lower false positive rate compared to other solutions investigated. In a study (Brijpal Singh, 2016), three algorithms, Min, Medium, and Max, were used for feature selection to identify anomalous intruders in the Internet of Things (IoT). The selected features were then applied to the SVM algorithm with RBF kernel for classification. The results of this study show that the proposed solution improves CPU execution time and detection performance.

Study (Sydney Mambue Kasongo, 2020) proposed a new intrusion detection model based on two-layer dimensionality reduction using LDA and PCA algorithms. The goal of this dimensionality reduction design is to detect disruptive activities such as user-to-root (U2R) and remote-to-local (R2L) attacks. The features selected for classification and detection of suspicious behavior were sent to two classification algorithms: Naive Bayes and k-Nearest Neighbors. The results show a significant improvement in the specific detection of U2R and R2L attacks compared to other models investigated.

Research (Hossein Shapoorifard, 2017) addresses severe network traffic imbalance across different layers and uneven distribution between training and test sets in the feature space through a multi-stage intrusion detection model called MSML. The MSML model introduces an improved K-Min algorithm called HSK-Mins, which normalizes the data through log-normalization.

The algorithm has limited flexibility and uneven distribution when selecting multiple parameters. The KDDCUP99 dataset was used to measure the performance of the MSML model. Empirical results show that MSML has a higher overall accuracy compared to other recognition models. Researchers in this study (Saad Mohamed Ali Mohamed Gadal, 2017) used deep learning theory and automatic feature extraction to improve the performance of an intrusion detection system. The method consists of a gated recurrent unit (GRU) recurrent neural network that selects the best features and classifies the results of a multi-layer perceptron (MLP) using a SoftMax module. The results show that it achieves lower false positive rates, higher accuracy, and good convergence speed. However, the detection of U2R and R2L attacks is still suboptimal.

Comparing this model with other models, it turns out that GRU is more suitable for IDS than Long Short-Term Memory (LSTM) units, which is an effective simplification and improvement of LSTM. Moreover, bidirectional GRU performs better than current solutions.

Research (Sheikhan, 2016) proposes a deep learning approach based on Self-Taught Learning (STL) based on Stacked Autoencoder (SAE). This strategy reduces the dimensions, testing, and training time while significantly improving the accuracy of the SVM algorithm in dealing with attacks. The results of this study include improved detection, reduced training and testing time, and reduced computational complexity of classification algorithms.

In one study (Hajisalem, 2018), researchers introduced a dimensionality reduction and feature selection technique by combining two algorithms: PCA and information gain. They combined three algorithms: SVM, IBK, and MLP to classify the selected features. In this study, the AOP algorithm was used to coordinate between these algorithms.

The study used datasets such as ISCX 2012, NSL-KDD, and KYOTO 2006+. The researchers found that using this method achieved the highest accuracy, lowest false positive rate, and improved computing performance. A study (Wang, 2017) proposed an intrusion detection system for deep learning networks based on SSAE. The study used the Min-Max algorithm to normalize the data and the LDS algorithm for dimensionality reduction. The researchers finally used a neural network with the SVM algorithm to classify the results. The dataset used in this study was NSL-KDD, and the results showed that although classification performance improved, it still had limited ability to detect U2R and R2L attacks.

In a study (Aslahi-Shahri, 2016), a multi-layer neural network based intrusion detection system was proposed. The dataset used in this study was KDD and a fuzzy C-means clustering approach was used to classify the inputs into different clusters. The results showed that while many previous studies implemented neural networks that could detect normal or attack connections, this study addressed a broader topic known as attack type detection.

Research (Guo, Chun et al., 2016) used an artificial neural network algorithm as a classifier using the KDD CUP 99 dataset to distinguish between normal and attack datasets through training and testing. In this study, MATLAB was used for simulation. The results showed that a feedforward neural network (FFNN) with 10 neurons and 2 layers performed better compared to FFNNs with different numbers of neurons but the same number of layers. In one study (Al-Yaseen, 2017), researchers investigated the detection of intrusions and attacks on computer networks using the deep learning algorithm FFDNN and feature extraction WFEU. This approach, known as WFEU-FFDNN, can be compared with other data mining algorithms such as SVM, RF, and NB, and according to the results of this study, it shows a higher accuracy than the studied methods. In one study (Nikam, 2015), the authors presented an intrusion detection system that uses a SSAE-based deep learning network, the Min-Max algorithm for data normalization, and the LDS algorithm for dimensionality reduction. They used the SVM algorithm to classify the results at the end of the neural network, which improved the classification performance, but showed limited ability to detect U2R and R2L attacks. In one study (T. G. Nguyen, 2019), three layers were implemented: edge, fog, and cloud, and each layer used a different classification algorithm. The SVM algorithm was used for the edge layer, SOM for

the fog layer, and SAE for the cloud layer. The combination of these algorithms optimized resources, solved the bottleneck problem, and ensured a good detection speed. However, a significant drawback is the network overhead.

Several machine learning algorithms, such as fuzzy logic, neural networks, support vector machines (SVM), and Naive Bayes (NB), have been used in the field of network intrusion detection (Mohamed Amine Ferrag, 2020). Each of these algorithms faces challenges such as computational complexity and long execution time, which can affect the accuracy of an intrusion detection system (IDS). An effective approach to improve intrusion detection is to select the best features.

It helps to identify useful features from IDS related datasets and eliminate suboptimal features. Feature selection methods used in conjunction with classification algorithms significantly improve the accuracy of IDS.

One of the latest advances in automated attack detection is the use of DL-PCA framework (Hariharan Rajadurai, 2020).

In this study, we employed a joint optimal selection technique of transfer function, SVM parameters, and feature selection using binary Grasshopper algorithm to increase detection accuracy and reduce error rate. The joint optimal selection of these parameters improves accuracy and provides a significant improvement compared to applying feature selection or parameter optimization alone.

THE PROPOSED METHOD

In this section, we present the proposed method for improving the accuracy of Intrusion Detection Systems (IDS) and identifying attacks using transfer functions and the binary Grasshopper algorithm. The proposed method is shown in Figure (1) and consists of three main phases: (see below)

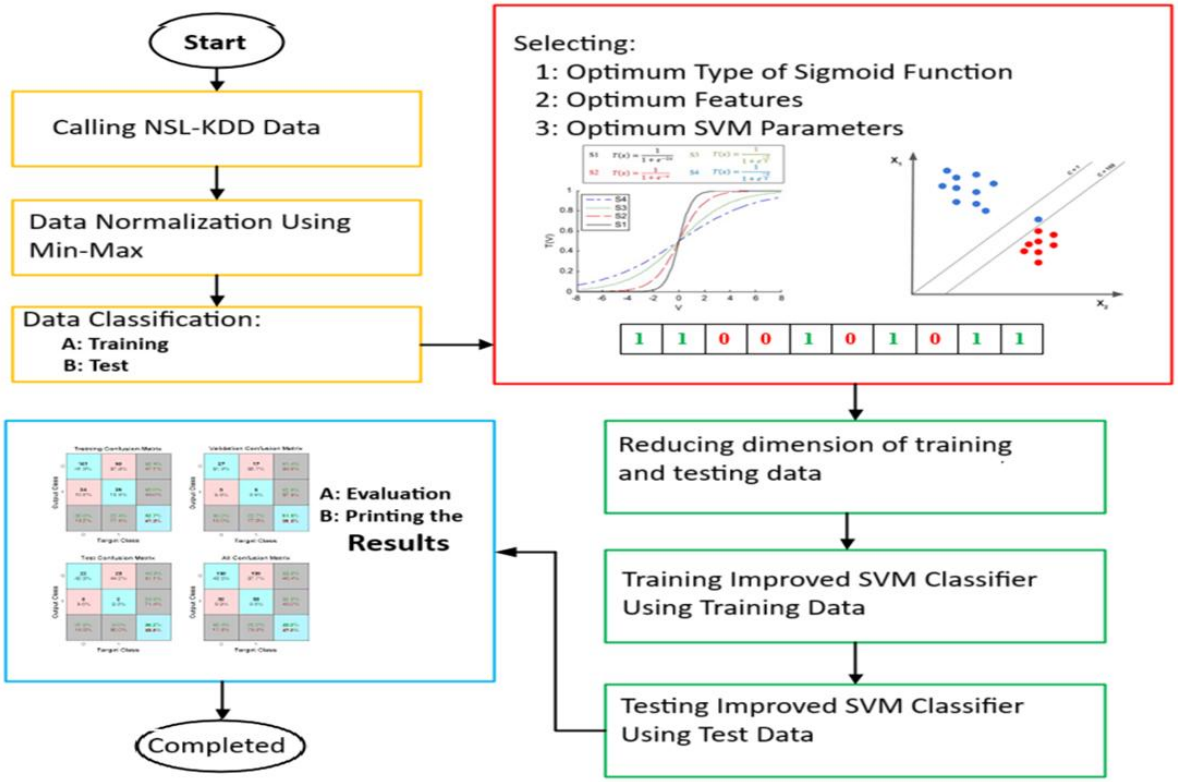

Figure (1) flowchart for the proposed method

Phase 1: Data Loading, Preprocessing, and Classification

The first phase of the proposed method involves preprocessing the data to improve the efficiency of classification. In this phase, we normalize the NSL-KDD dataset and convert the data values of each column into the range of 0 to 1. Such normalization ensures that the data falls within a certain range, which leads to better results. In this study, we perform normalization using the min-max method, which is expressed by Equation (1).

1. $$Z = \frac{X - \min(x)}{\max(x) - \min(x)}$$

In this formula, $\boldsymbol{X}$ is the value to be normalized, min(x) is the minimum number in the data set, and max(x) is the maximum number.

S-shaped family		V-shaped family	
Name	Transfer function	Name	Transfer function
S1	$T(x)=\frac{1}{1+e^{-2x}}$	V1	$T(x)=\left\|\operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x\right)\right\|=\left\|\frac{\sqrt{2}}{\pi}\int_0^{(\sqrt{\pi}/2)x}e^{-t^2}dt\right\|$
S2 [8]	$T(x)=\frac{1}{1+e^{-x}}$	V2 [13]	$T(x)=\left\|\tanh(x)\right\|$
S3	$T(x)=\frac{1}{1+e^{(-x/2)}}$	V3	$T(x)=\left\|(x)/\sqrt{1+x^2}\right\|$
S4	$T(x)=\frac{1}{1+e^{(-x/3)}}$	V4	$T(x)=\left\|\frac{2}{\pi}\operatorname{arc}\tan\left(\frac{\pi}{2}x\right)\right\|$

Standard Grasshopper Algorithm

Grasshopper belong to the insect family and are considered pests due to their ability to cause damage to crops and agriculture. To model the behavior of grasshopper, equation (2) is used. The data is then split into two groups: training (constituting 80% of the data) and testing (constituting 20% of the data).

Phase 2: Sigmoid function selection and feature selection with BGOA algorithm

In this phase, the optimal sigmoid function is selected. A sigmoid function is a type of transfer function that defines the probability that an element of a position vector changes from 0 to 1 or from 1 to 0. Here are some concepts to consider in the transfer function:

1. Since particles may need to change position, the range of the transfer function should be restricted to the interval [0,1].

2. The transfer function should provide a high probability of a position change in the value of the velocity vector.

3. The transfer function should provide a low probability of a position change in the value of the velocity vector.

4. The return value of the transfer function should increase simultaneously with an increase in velocity. Particles moving away from the optimal solution have a higher probability of changing their position vector and returning to their previous position.

5. The return value of the transfer function should decrease simultaneously with the velocity.

Figure (2) shows the transfer function used in the proposed method (Mirjalili, S-Shaped and V-Shaped Transfer Functions for Binary Particle Swarm Optimization, 2013).

After selecting the sigmoid function, the binary Grasshopper algorithm is used to select the optimal features.

2. $X_i = S_i + G_i + A_i$

In equation (2), x represents the location of the locust, Si indicates the social interaction between locusts, G represents the gravity acting on the locust, and A indicates the wind direction. These three elements represent the location of the locust and can be used to generate a random behavior using equation (3), where r varies randomly between 0 and 1. (Yaghobzadeh, A Binary Locust Optimization Algorithm for Feature Selection, 2020)

3. $X_i = r_1 S_i + r_2 G_i + r_3 A_i$

The value of Si, calculated according to equation (4), represents the social interaction: where dij denotes the distance between grasshoppers i and j.

4. $S_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} s\,(d_{ij})\widehat{d_{ij}}$

In equation (4), dij represents the distance between grasshoppers i and j, which is calculated as follows: The function s acts as a map of the distance between grasshoppers and is derived according to equation (5).

5. $s\,(r) = fe^{\frac{-r}{l}} - e^{-r}$

In this equation, f indicates the strength of the attractive force. The general formula is expressed in equation (6).

6. $X_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} s\,(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - g\widehat{e_g} + u\widehat{e_w}$

In this equation, N represents the number of grasshoppers. However, since the grasshoppers move on the ground, their positions should not exceed a certain threshold. Therefore, the modified equation (7) is used.

7. $X_i^d = c\left(\sum_{\substack{j=1 \\ j\neq i}}^{N} c\frac{ub_d - lb_d}{2} s\,(|x_j^d - x_i^d|)\frac{x_j - x_i}{d_{ij}}\right) + \widehat{T_d}$

The parameter C is one of the important parameters of the locust optimization algorithm. It is a reduction coefficient that affects the discussion of the safe area, repulsion area, and attractive area. This parameter update is given by the following relationship (8):

8. $c = cmax - l\frac{cmax-cmin}{L}$

Binary Grasshopper Algorithm

In the standard Grasshopper algorithm, the solution is updated towards a continuous value. For feature selection in this algorithm, the search space is modeled as a Boolean network. The solution to the problem of feature selection or non-selection is to use a binary vector. This algorithm contains a vector of 0s and 1s, where 1 indicates feature selection and 0 indicates feature non-selection.

This algorithm is transformed into a binary Grasshopper algorithm in three main steps:

1 : Update of parameters c, 2: Nonlinear mapping v, and 3: Update of the new Grasshopper position. To build this algorithm, modified equations (9), (10) and (11) are used. In equation (9), instead of G+A, a designation T is used, which corresponds to the best grasshopper found so far. Equation (10) describes the social interactions of the grasshoppers.

Equation (9): T represents the best grasshopper found so far.

Equation (10): describes the social interactions between the grasshoppers.

The effectiveness of the algorithm depends on these interactions and updates to ensure optimal performance in the feature selection task.

9. $X_i^d = c\, s_i + T_d$

Because the decision variables in the binary grasshopper algorithm are only between 0 and 1, we use a nonlinear function called V that maps the grasshopper's social behavior into another function or space. This value is calculated using equation (10). Using this nonlinear mapping allows us to more effectively represent the interactions between the grasshoppers, thereby allowing the algorithm to better explore the search space while maintaining a balance between exploration and exploitation. This approach improves the algorithm's ability to find optimal solutions to a variety of optimization problems.

10. $V_i = \left|\frac{2}{\pi}\ tan^{-1}\left(\frac{\pi}{2}\ c\ s_i\right)\right|$

Finally, the position of the grasshopper obtained by modifying and adjusting the formula needs to be updated. This update is calculated according to equation (11).

11. $\begin{cases} T_i & p > v_i \\ X_i & p < v_i \end{cases} \quad X_i$

The proposed formula can perform exploration and exploitation in the search space. However, a mechanism is needed to adjust the level of exploration and exploitation of the search agent. The reason for using simultaneous exploration and exploitation in this algorithm is to initially maintain a global perspective on the search space, consider all possible solutions, and then switch to a local perspective towards the end of the algorithm's execution to find the optimal solution. Search.

Phase 3: Classification of the selected data using the optimized SVM algorithm

The SVM classification method, which is one of the linear classification methods, tries to find the optimal hyperplane with the maximum margin to separate the data into two classes. The training data is represented by pairs (xi, yi), where xi are n-dimensional features and yi ∈ {−1,1}. The goal is to find a robust model that can separate two classes labeled -1 and 1 with maximum distance. As shown in figure (2), a strong separator y has a maximum separation limit. The vertical distance above the points is obtained by dividing the absolute value of the parameter bb by the norm of w. The main idea is to choose a suitable separator that maximizes the distance of each class to its neighbors. This solution is actually the one that has the maximum distance from the points of the two classes, and the boundary can be separated by two parallel hyperplanes that pass through at least one point of each class. These vectors are called support vectors. The mathematical formula for these two parallel hyperplanes that form the separating boundary is shown in equation (12).

12. W.x-b=1 W.x-b=-1

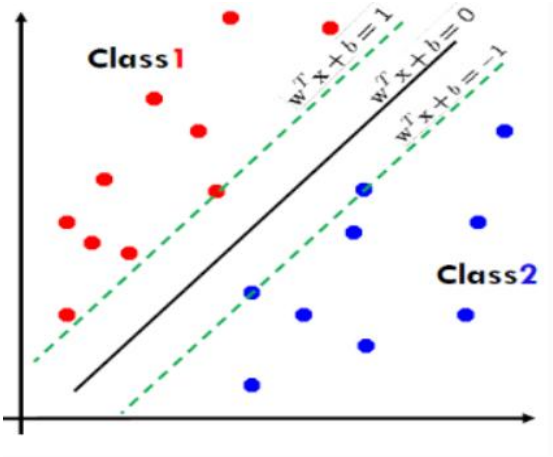

Figure 2: Support Vector Machine

It should be considered that if the training data is not linearly separable, two parallel hyperplanes can be chosen such that there are no data points between them. The distance between these two parallel hyperplanes needs to be maximized. According to the geometric theorem (13), this distance is equal to |w|/2, so the total distance is |*w*|.

13. W .x-b≥1 W. x-b≤-1

This constraint is shown in the following equation (14):

Ci (w. x-b) =1 1 ≤ i ≤-1 (17)

RBF Kernel: The most common kernel type in SVM algorithms is the Radial Basis Function (RBF) kernel, which is popular because it has a localized and limited response across the full range of the vector X.

For $-\gamma > 0$, the following parameter is sometimes used:

The formula is as follow:

$$k(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

σ is a free parameter and represents the width of the RBF kernel function.

$$k(x,y) = exp(-\gamma \|x-y\|^2$$

To simulate and evaluate the method proposed in this study, MATLAB is used as a computing environment for numerical calculations and simulations related to data mining and artificial intelligence algorithms.

FINDINGS

To simulate, test, and determine the effectiveness of an intrusion detection system, a standard data set must be used. The first standard dataset, KDDCUP99, was collected over several weeks by the IST group at MIT Labs under the supervision of DARPA and AFRL/SNHS. This dataset contains 41 features per data point. In a unit set of 41 features, an observation can represent either an attack or a normal state. The training data contains 26 known attack types and the test data contains 14 unknown attack types and is often used to evaluate unmonitored intrusion detection systems. The dataset is divided into normal traffic and four attack classes.

Dataset Name	Number of Classes	Number of Features	Class 1	Class 2		Class 3	Class 4	Class 5
NSL-KDD	5	41	Normal	R2L		U2R	Probe	Dos

Comparing the results of the accuracy metrics of the proposed method, we can see that the sigmoid function T(x) is a sigmoid function. Figure (3) shows the comparison of the accuracy metrics of the proposed method and the reference paper.

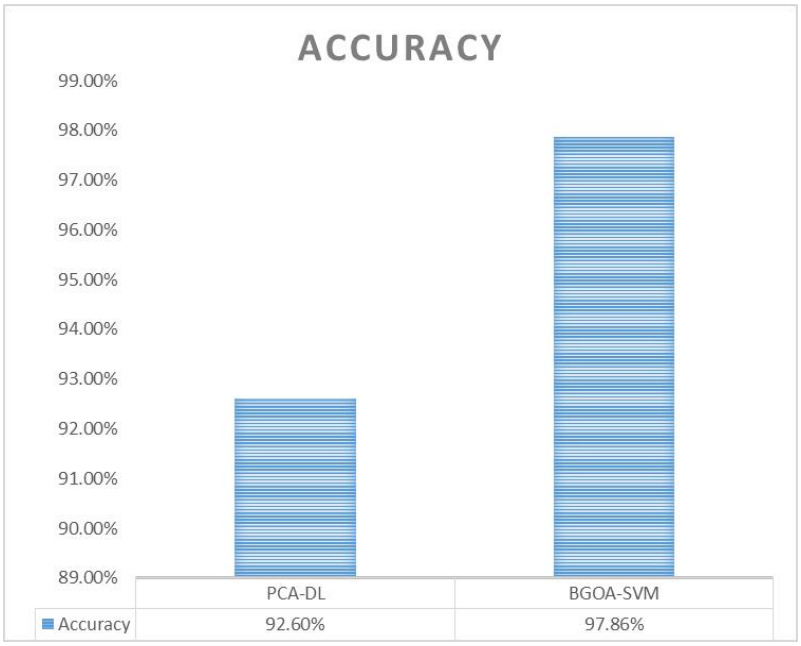

Figure (3) comparison of the accuracy metrics of the proposed method and the reference paper

Figure (3) shows a comparison of the accuracy metrics of the proposed method and the reference paper (Hariharan Rajadurai, 2020). From the obtained results, it can be seen that the proposed method has a higher evaluation ability than the reference paper. This indicates that the improvement of the support vector machine algorithm and the use of the sigmoid function in the Binary Grasshopper algorithm, as well as the identification of important and influential features in intrusion detection, have led to an improvement and increase in the detection rate. Unlike other algorithms, the Whale algorithm takes into account the concepts of both exploration and exploitation, and therefore is not trapped in local optimization. A notable feature of the Grasshopper algorithm compared to other algorithms such as PSO is that in the Grasshopper algorithm, there is only one position vector for each search agent. Furthermore, the Grasshopper algorithm updates the search agent based on its current position, the best global solution, and the positions of other Grasshoppers. The evaluation of the criteria considered in the proposed method with and without feature selection using the binary Grasshopper algorithm is shown in Table (2).

F-measure (%)	Detection (%)	Recall (%)	Accuracy (%)	Evaluation Metrics
97.42	96.67	77.66	96.66	With feature selection
92.69	88.62	82.03	95.56	Without feature selection

Table 2: Evaluation of Accuracy, Recall, Detection, and F-measure metrics.

As shown in table (2), the use of feature selection with the sigmoid function and the Grasshopper algorithm improved the evaluation metrics, since the best features were selected.

Feature selection may reduce computational complexity (both in terms of execution load and memory). On the other hand, dimensionality reduction also contributes to a deeper understanding of the data. This issue is particularly important in many applications (e.g. classification), where there is a large number of features, many of which are not used or do not carry much information weight. Not removing these features does not cause information problems, but it increases the computational load of the desired application. In addition to useful data, a lot of non-useful information is also stored.

The proposed method of selecting a subset of features from the NSL-KDD dataset using the Binary Grasshopper algorithm identified the following features as more important for an Intrusion Detection System (IDS) (Table 3):

Sample Data	Feature Description	Feature Name	Identifier
SF	Connection status - normal or error	Flag	F4
Urgent	Number of bytes transferred from source to destination in a single connection	Src_bytes	F5
0	If the source and destination IP addresses and port numbers are the same, this variable takes the value 1; otherwise, it takes the value 0.	Land	F7
0	Total number of erroneous segments in this connection	Wrong_fragment	F8

0	Number of urgent packets in this connection. Urgent packets are those with the urgent bit set.	Urgent	F9
0	Number of "Hot" indicators in the content, such as: login attempts.	Hot	F10

Table 3: Selected Features

CONCLUSION

The primary obstacle encountered in computer networking is the prevalence of attacks, which can occur regardless of the network's scale. A computer network attack involves unauthorized individuals gaining illegal access to servers or other hardware and software resources. In response, various companies have developed strategies to combat these threats. Notably effective techniques recently implemented include Support Vector Machines (SVM) and k-nearest neighbor classification. However, achieving optimal detection accuracy requires careful selection of criteria when applying these methods.

This study utilized MATLAB as the computational environment for numerical analysis and simulations related to data mining and artificial intelligence algorithms. To enhance detection accuracy and minimize error rates, a combined optimization approach was employed for the transfer function, SVM parameters, and feature selection using the Binary Grasshopper algorithm. This simultaneous optimization of parameters significantly boosts accuracy compared to scenarios where only feature selection or parameter optimization is conducted. The findings indicate a 5.26% increase in accuracy over the reference study.

Based on these results, it is recommended to adjust the objective function for feature selection, potentially by incorporating metrics such as root mean square error (RMSE) and normalized root mean square error (NRSME), or by expanding the dataset with additional classes.

REFERENCES

Al-Yaseen, W. L. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Systems with Applications, 296-303.

Aslahi-Shahri, B. M. (2016). A hybrid method consisting of GA and SVM for intrusion detection system. Neural Computing and Applications, 1669-1676.

Binghao Yan, G. H. (2018). Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE Access, 41238-41248.

Brijpal Singh, A. K. (2016). Innovative Empirical Approach for Intrusion Detection Using ANN. International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 94-101.

F. Salo, A. B. (2019). Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. Computer Networks, 164-175.

Guo, C. e. (2016). A two-level hybrid approach for intrusion detection. Neurocomputing, 391-400.

Guo, Chun, et al. (2016). A two-level hybrid approach for intrusion detection. Neurocomputing, 391-400.

Hajisalem, V. a. (2018). A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. Computer Networks, 37-50.

Hariharan Rajadurai, U. D. (2020). An empirical model in intrusion detection systems using principal component analysis and deep learning models. Computational Intelligence, 1-12.

Hossein Shapoorifard, P. S. (2017). Intrusion detection using a novel hybrid method incorporating an improved KNN. International Journal of Computer Applications, 5-9.

Mirjalili, S. &. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation, 1-14.

Mirjalili, S. &. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation, 1-14.

Mirjalili, S. &. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation, 1-14.

Mirjalili, S. &. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation, 1-14.

Mohamed Amine Ferrag, L. M. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 102419-102434.

Mohammed A. Ambusaidi, X. H. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Transactions on Computers, 2986-2998.

Nikam, S. S. (2015). A comparative study of classification techniques in data mining algorithms. Oriental Journal of Computer Science & Technology, 13-19.

Saad Mohamed Ali Mohamed Gadal, R. A. (2017). Anomaly detection approach using hybrid algorithm of data mining technique. 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE 2017) (pp. 1-6). Khartoum: EEE.

Sheikhan, M. a. (2016). A hybrid intrusion detection architecture for internet of things. 2016 8th International Symposium on Telecommunications (IST), 601-606.

Sunita S., B. J. (2016). A Hybrid approach of Intrusion Detection using ANN and FCM. European Journal of Advances in Engineering and Technology, 6-14.

Sunita S., B. J. (2016). A Hybrid approach of Intrusion Detection using ANN and FCM. European Journal of Advances in Engineering and Technology, 6-14.

Sydney Mambwe Kasongo, Y. S. (2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Computers & Security, 101752-101763.

T. G. Nguyen, T. V.-I. (2019). SeArch: A Collaborative and Intelligent NIDS Architecture for SDN-Based Cloud IoT Networks. IEEE Access, 127272–127290.

Wang, H. J. (2017). An effective intrusion detection framework based on SVM with feature augmentation. Knowledge-Based Systems, 130-139.

Yaghobzadeh, R. K. (2020). A Binary Grasshopper Optimization Algorithm for Feature Selection. International Journal of Engineering Research & Technology (IJERT), 533-540.

Yaghobzadeh, R. K. (n.d.). A Binary Grasshopper Optimization Algorithm for Feature Selection.