International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

DDoS Attack Detection using Swarm Optimized Random Forest Classification

R. Sarath Babu*[1], Dr. K. Radhika[2]

Submitted: 11/10/2023 **Revised**: 30/11/2023 **Accepted**: 11/12/2023

Abstract: With the increasing spread of the Internet, the need for security also increases - both in the private and in the business sector. Corporate networks in particular are often exposed to attempted attacks. In order to avert or limit the damage, these attacks must be recognized and appropriate countermeasures initiated. This task is achieved by an Intruder Detection System (IDS). This paper presents a DDoS attack detection model using swarm optimization-based feature selection and Radom Forest (RF) classifier. A modified Grey Wolf Optimization (GWO) algorithm is used to select the features which produce the best accuracy. The fitness function of the conventional GWO algorithm is replaced with Stochastic Gradient Descent (SGD) in order to perform feature selection. The RF classifier is then trained using the chosen subset of features to identify attacks. The proposed model is tested on CICIDS2017 dataset and has been compared with existing machine learning techniques to evaluate the efficiency of the proposed model. GWO earned the highest Accuracy of 99.8. This was accomplished with only 40 out of 75 features. When GWO provided the least number of features, 38, resulting in accuracy of 99.7. Over several experiments, modified GWO with DT had an average classification accuracy of 99.5 percent.

Keywords*: DDoS, Grey Wolf Optimization (GWO), Intruder Detection System (IDS), Radom Forest (RF), Stochastic Gradient Descent (SGD).*

1. Introduction

Intrusion detection is the active monitoring of computer systems and / or networks with the Target of attack and abuse detection identified [1]. The goal of intrusion detection is in filtering out all of the events taking place in the surveillance area Attacks, attempts to abuse or security breaches indicate, then deepened to investigate. Events should be recognized and reported promptly. Intrusion detection is to be understood as a process and requires suitable organizational integration as well as technical support using suitable tools [2].

IDS are used to detect attacks on the computer network. A distinction is made between IDS on the one hand based on their detection method (anomaly-based vs. signature-based) and on the other hand based on their area of responsibility (host-based vs. network-based). Intrusion detection system is a set of tools that covers the entire intrusion detection process, from event detection to evaluation and escalation. Support and documentation of events. The majority of Intrusion Detection products available on the market have this integrated functionality. However, IDS can also be composed of individual components. The selection and compilation of the IDS are based on this the individual technical and organizational conditions and requirements.

Today, it should come as no surprise that one of the primary goals of an IDS is to ensure the safety of a computer, a network, or both. There is seldom a day that goes by in which one does not hear about fresh attempts to crack business networks or hack into such networks. As a result of attempted burglaries, some credit card data have already been obtained by dishonest persons who are criminals.

In the most recent few years, both the total number of assaults and the expenses that are linked with them have skyrocketed, as indicated by data obtained from the Computer Emergency Response Team (CERT). This tendency could not be fought against at this time; even so-called intrusion prevention technologies frequently failed to accomplish their goal. There are many different ways that danger might present itself. For instance, the attacker may have obtained access to the system or its resources by exploiting flaws in the manner in which the TCP/IP stack was implemented. If a trained expert is able to determine which system is being used based on an analysis of the TCP/IP fingerprint (nmap), they are then able to utilize this knowledge to look for certain vulnerabilities and eventually exploit such vulnerabilities during an attack.

In addition, vulnerabilities may frequently be identified in the software that is installed on the system. These vulnerabilities typically take the form of buffer overflows, which a typical user is able to exploit. You even have the chance to obtain root rights. The DNS daemon BIND, which has been the market leader under Unix and its variants for decades, the FTP server WuFTP, and of course the tried-and-true Microsoft Internet Information Server all

[1] *Research Scholar, Department of Computer Science and Engineering, University College of Engineering, Osmania University, Hyderabad, Telangana, India.*
[2] *Professor and Head, Department of Information Technology, Gandipet, Hyderabad, Telangana, India. Email: kradhika_it@cbit.ac.in*
** Corresponding Author Email: sharath.rakki@gmail.com*

performed extremely poorly. The first two programs were responsible for introducing worms into the Unix environment. There is currently a worm that exploits many vulnerabilities in the Microsoft Internet Information Services (IIS) and BIND web servers [3].

It is a common misconception that attempts to breach security are exclusively carried out within the context of corporate networks or huge organizations. In point of fact, every single person who uses the private internet is vulnerable to assault. However, an IP address, specifically the one that is known to frequently change behind plug-end computer systems, is not even close to as interesting as a system that maintains permanent contact with the Internet or multiple systems that are connected together in a network, particularly in light of the fact that distributed denial of service attacks used to be possible. A computer that connects to the Internet using flat rate DSL is far more vulnerable to attack than a machine that connects to the Internet using ISDN or an older modem only occasionally.

In the end, a chain is only as strong as its weakest link, which in this case is the individual who is using the system. Passwords that are too easy to guess are frequently the cause of attempted break-ins. Once a criminal has gained access to the system, it is simple for them to obtain even greater levels of privileges, or even absolute root rights, on a computer. In addition to attempts using what are known as brute force, the attacker can, of course, also attempt to steal the user's password by using social engineering. There are always going to be instances in which unscrupulous individuals try to coerce a user into divulging their password under the guise of a pretext or a false identity. This kind of attack (using passwords) can't be stopped by an IDS, of course. One can, at most, observe odd behaviors that come from this login, but by that point, it is sadly frequently too late to take action.

In the absence of an IDS, it becomes difficult to determine how long an intruder was able to operate undetected, how the intruder carried out his attack, or how much damage was caused. In a nutshell, the following points can sum up the objectives that are intended to be achieved through ID systems:

- notifying those responsible (administrator, security officer) or taking active countermeasures in the case of an attack,
- the legal applicability of the information that was gathered;
- The detection of a loss of data,
- Protection from potential future assaults by analyzing the information gathered in the case of a (simulated) break-in

In [4], the authors proposed a network-IDS that reads all of the packets and investigate them for unusual patterns. In order for IDS to be able to accommodate the large bandwidths of today's networks, it must enable high performance while processing and analyzing data. I t is impossible to guarantee that the IDS will do complete monitoring. Hybrid IDS combines host-based and network-based security measures to provide even greater levels of protection.

A comprehensive analysis of the scientific work that has been done on the detection of intrusions using machine learning approaches during the past ten years is presented in [5]. In addition to that, the authors discussed about a few of the unresolved problems that are still outstanding. This survey acted as a supplementary that have been conducted before on the topic of intrusion detection and is designed to support earlier research. Additionally, it will provide researchers who are working on intrusion detection using ML algorithms with a ready reference for their work.

Uhm and Pak [6] presented a fresh approach that makes use of service-aware dataset partitioning, which not only offers high scalability to manage large amounts of data that are expanding at a rapid rate in a flexible manner but also assists the classifier in improving their speed and accuracy. The authors assessed the method using the Kyoto2016 dataset, a dataset for severely unbalanced data.

Zhou et al. [7] presented a framework for the detection of intrusions, based on the approaches of feature selection and ensemble learning. In the first stage of dimensionality reduction, a heuristic approach known as CFS-BA is developed. This algorithm finds the ideal subset by basing its decisions on the association between the features. Then, the authors provide an ensemble method that is a combination of the C4.5 algorithm and Random Forest (RF). In the end, a voting mechanism is utilized to integrate the probability distributions of the several base learners in order to recognize attacks.

Because of the high volume of traffic, an IDS performs analysis on large amounts of data, and it also safeguards data and computer networks from hostile activity. Therefore, in order to differentiate between typical and suspicious activity, a classification method that is both quick and effective is necessary. There are a variety of approaches that make use of the machine learning methodology, and these approaches have recently been available for use in intrusion detection. In [8], a variety of IDS strategies that are based on machine learning are examined, explained, and categorized. IoT is one of the upcoming internet technologies that focuses on the delivery of services and adjusting the way that technologies are implemented across various communication networks [15-17].

2. Proposed Model

The proposed model takes the input dataset containing the network parameters as input. These features are given to the modified GWO model for feature selection. The selected features are used to train the random forest model, which later is used to classify the test data. Fig. 1 shows the proposed classification model.

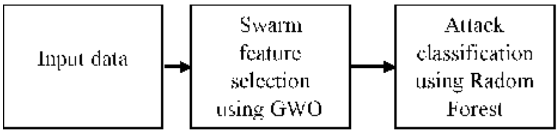
Input data
Swarm feature selection using GWO
Attack classification using Radom Forest

Fig. 1. Proposed Attack Classification Model

2.1. Modified Grey Wolf Optimization (GWO)

This optimization technique is based on the pattern of hunting pattern of the grey wolves [9]. The wolves in the search proves are labelled as alpha, beta, delta, and omega. The process of finding the optimal solution is divided into the following phases: searching, encircling and attacking. The algorithm of GWO is presented here:

```
$Initialize\ the\ grey\ wolf\ population\ Xi(i$
            $= 1, 2, \ldots, n)\ with\ NN\ weights$
$Initialize\ a, \vec{A}, and\ \vec{C}, where\ \vec{A}$and $\vec{C}$are coefficient vectors, an
$\in [0,2]$
$Calculate\ the\ fitness\ of\ each\ search\ agent$
$\vec{X}_\alpha\ =\ the\ best\ search\ agent$
$\vec{X}_\beta = the\ second\ best\ search\ agent$
$\vec{X}_\delta\ =\ the\ third\ best\ search\ agent$
$while\ (t < Maxnumber\ of\ iterations)$
$for\ each\ search\ agent$
    $Update\ the\ position\ of\ the\ current\ search\ agent$
$endfor$
$Update\ \alpha\ , A, and\ C$
$Calculate\ the\ fitness\ of\ all\ search\ agents$
$Update\ X_\alpha, X_\beta, and\ X\_\delta$
$t = t + 1$
$end\ while$
$return\ X\_\alpha$
```

The fitness function used in GWO is stochastic gradient boost.

2.2. Stochastic Gradient Descent

The idea of shifting parameters is crucial to understanding the gradient approach. Using linear regression as an example, this idea is better conveyed. Here, a single variable's linear regression is discussed [10-14]. A fixed number of points make up a linear regression with one variable.

$$(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$$

A straight line that reduces the sum of errors $y = f(x)$is utilized to approximate the points. For example, in the below fig. 2, the solution is red coloured straight line.

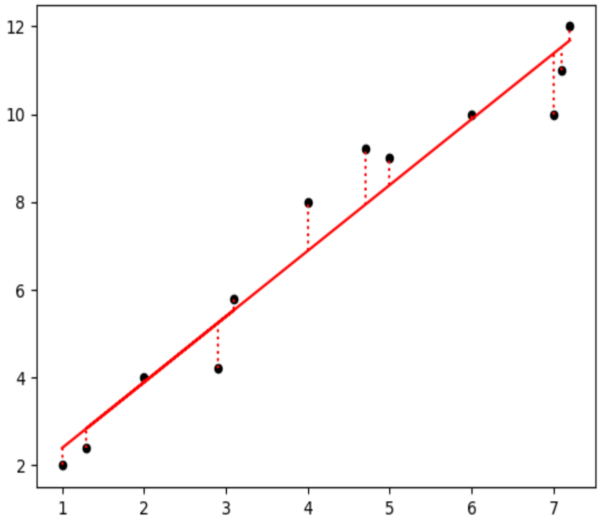

Fig. 2. SGD Classifier

Typically, the sum of squares of residuals is used to compute the errors of a straight line and several points. The following formula provides it in more detail.

$$sum\ of\ squares\ of\ residuals = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Now, $y = f(x)$ is a straight line, which can be represented as $f(x) = ax + b$, where a denotes the slope of a line and b denotes an intercept. The line is moved using the parameters a and b. When a parameter is changed, the goal function's error value also changes. Utilizing the connection in between the parameter and goal function, the gradient descent approach minimizes error. Only one randomly chosen data is utilized for each parameter update in SGD, which is its fundamental working principle. In other words, utilizing just one data rather than all the data for each parameter change significantly reduces the amount of work. The distance between the straight line that corresponds to the current parameters (the red straight line in the picture below) and one randomly chosen point is the sole factor taken into account by SGD while updating the parameters.

The steps listed below are used to implement the SGD algorithm:

Step 1: Each parameter's relationship to the input data's slope/gradient is determined.

Step 2: Calculate the partial derivative of the output with respect to each input parameter using a random set of input parameters.

Step 3: By altering the set size, the gradient function will be updated.

$$step\ size\ =\ gradient\ *\ learning\ rate$$

Step 4: search for the new parameters:

$$new\ params\ =\ old\ params\ -step\ size$$

Step 5: Repeat steps 2–4 until the gradient is nearly zero.

2.3. Random Forest (RF) Classifier

Random Forest (RF) is a supervised machine learning method used for data classification and regression applications. It is based on the concept of ensemble learning, a technique of merging several different classifiers into a single system in order to tackle challenging problems and improve the model's overall effectiveness. RF is a classification system that utilizes a large number of decision trees on various subsets of the given dataset and then averages the results in order to improve the accuracy of the predictions made using the dataset as a whole. The RF does not rely on a single decision tree but instead takes forecasts from all of the trees and then predicts the outcome based on the majority of those projections. Because there are more trees in the forest, there is less of a chance of the model being overfit, which results in greater accuracy.

The RF algorithm, like any other ML technique, has two phases. In the training phase, the Decision Tree (DT) subsets are created and combined into a large RF. These trees are later used to test new data for classification and regression purposes. This paper used RF algorithm to classify the attacks happening on a network. The steps involved in the algorithm are as follows:

Step 1: Select a random subset of points from the input dataset.

Step 2: Construct DT for the selected subset of data points.

Step 3: Evaluate the performance of the selected DT

Step 4: Repeat steps 2 and 3 to reach the desired accuracy.

Step 5: For test data, find the results of the DTs and classify the data.

Decision Tree (DT)

DT are frequently constructed to match the way in which people think while they are making decisions, which makes them easy to understand. The logic that underpins the decision tree is intuitively clear because to the fact that it is organized in a tree-like fashion. When working with a DT, the algorithm starts with a root node and it moves towards predicting the category of the provided dataset. This method navigates down the branch and advances to the next node by comparing with the reference value. Before moving on to the next node, the algorithm does a last check to ensure that the attribute value is consistent with that of the previous sub-nodes. The method is carried out until the tree's leaf node is reached.

- Root node: The starting node of the DT.
- Leaf node: The final node without any branch.
- Decision node: Intermediate nodes between root node and leaf node.
- Splitting: The phenomena that occurs after the decision node.

Sub-Tree: The branch network of the DT.

These terms are illustrated with the help of fig. 3.

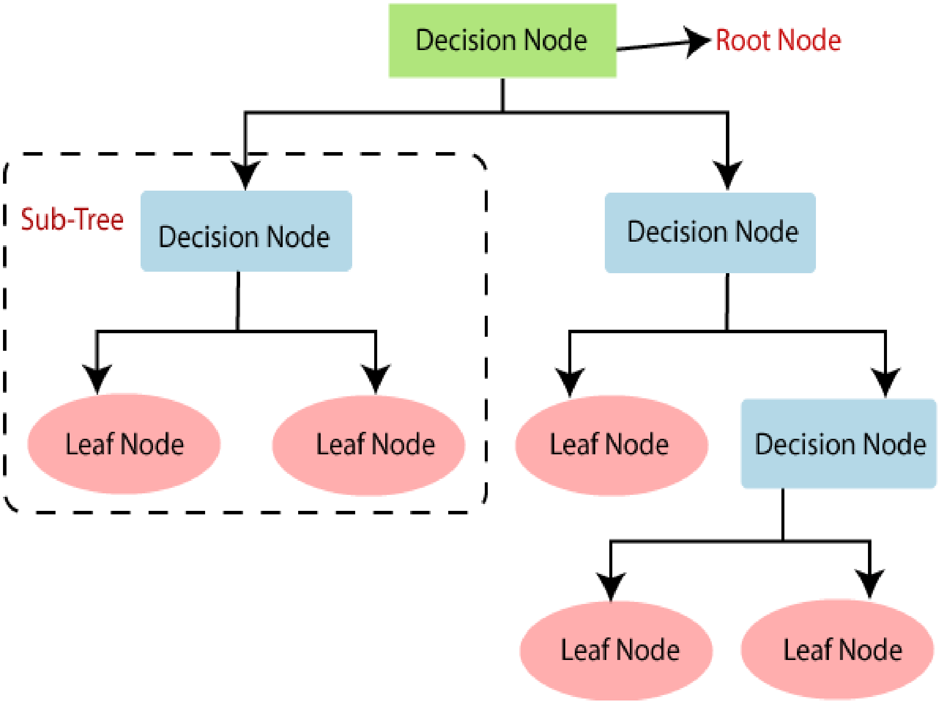

Fig. 3. DT Structure

The algorithm for the DT is as follows:

Step 1: Read the input data features at the root node.

Step 2: Convert the root node into a decision note and add a condition to the decision node which results in two leaf nodes.

Step 3: Select the left leaf node and transform it into a decision node by adding a condition. Repeat the same thing with the right node.

Step 4: Perform step 3 repeatedly by updating the conditions in the decision nodes until the leaf nodes classify the input data.

Step 5: Save the network structure with the conditions in the decision nodes.

In the experimental analysis, the performance of RF is evaluated on a dataset and compared with other classification techniques. The next section presents a detailed description of the experimental results carried out.

The proposed model is tested on CICIDS2017 dataset. The total number of entries in the dataset is 3000, of which 1500 samples are attacks and 1500 samples are genuine user requests. The total features in the input dataset are 78. The modified GWO algorithm is used to select the features in the dataset and the selected subset of features is classified using RF algorithm. To validate the proposed model, different optimization algorithms are first tested, which include Particle Swarm Optimization (PSO), Multi-Verse Optimizer (MVO) and GWO. Table 1 shows the training results of different classification techniques when using PSO for feature selection. Over a set of 5 different experiments, SGD obtained a maximum classification accuracy of 93.9% and an average classification accuracy of 90.8%. SVM obtained a maximum classification accuracy of 93.3% and an average classification accuracy of 92.98%. MLP obtained a maximum classification accuracy of 99.6% and an average classification accuracy of 99.4%. DT obtained an accuracy of 100% in all the experiments.

3. Experimental Results

Table 1. PSO training results with different machine learning techniques

Experiment	*SGD*	*SVM*	*MLP*	*DT*
1	0.939868621	0.927741283	0.996968166	1
2	0.887822132	0.928751895	0.995452249	1
3	0.938352703	0.928246589	0.993431026	1
4	0.84689237	0.931278423	0.994946943	1
5	0.927741283	0.933299646	0.98938858	1

Table 2 shows the testing results of different classification techniques when using PSO for feature selection. SGD obtained a maximum classification accuracy of 93.63% and an average classification accuracy of 93.38%. SVM obtained a maximum classification accuracy of 94.31% and an average classification accuracy of 94.12%. MLP obtained a maximum classification accuracy of 99.5% and an average classification accuracy of 99.2%. DT obtained a maximum accuracy of 99.7% and an average accuracy 99.6%.

Table 2. PSO testing results with different machine learning techniques

Experiment	*SGD*	*SVM*	*MLP*	*DT*
1	0.934378061	0.940254652	0.995103	0.996082
2	0.887365328	0.942213516	0.990206	0.995103
3	0.936336925	0.93927522	0.993144	0.995103
4	0.826640548	0.941234084	0.994123	0.997062
5	0.934378061	0.943192948	0.988247	0.997062

Table 3 shows the training results of different classification techniques when using MVO for feature selection. SGD obtained a maximum classification accuracy of 93.27%. SVM obtained a maximum classification accuracy of 92.9%. MLP obtained a maximum classification accuracy of 99.7%. DT obtained an accuracy of 100% in all the experiments.

Table 3. MVO training results with different machine learning techniques

Experiment	*SGD*	*SVM*	*MLP*	*DT*
1	0.900960081	0.928246589	0.991409803	1
2	0.931278423	0.927741283	0.99646286	1
3	0.924204144	0.929257201	0.990904497	1
4	0.914098029	0.927235978	0.997978777	1
5	0.932794341	0.909044972	0.974229409	1

Table 4 shows the testing results of different classification techniques when using MVO for feature selection. SGD obtained a maximum classification accuracy of 93.04%. SVM obtained a maximum classification accuracy of 94.22%. MLP obtained a maximum classification accuracy of 99.6%. DT obtained a maximum accuracy of 99.6%.

Table 4. MVO testing results with different machine learning techniques

Experiment	*SGD*	*SVM*	*MLP*	*DT*
1	0.914789422	0.940254652	0.990206	0.994123
2	0.930460333	0.941234084	0.992165	0.995103
3	0.917727718	0.942213516	0.994123	0.995103
4	0.926542605	0.940254652	0.996082	0.994123
5	0.929480901	0.922624878	0.975514	0.996082

Table 5 shows the training results of different classification techniques when using modified GWO for feature selection. Over a set of 5 different experiments, SGD obtained a maximum classification accuracy of 93.07% and an average classification accuracy of 91.84%. SVM obtained a maximum classification accuracy of 93.48% and an average classification accuracy of 82.15%. MLP obtained a maximum classification accuracy of 99.6% and an average classification accuracy of 97.02%. DT obtained an accuracy of 100% in all the experiments.

Table 5. Modified GWO training results with different machine learning techniques

Experiment	*SGD*	*SVM*	*MLP*	*DT*
1	0.859019707	0.656897423	0.991915109	1
2	0.481556342	0.663466397	0.902981304	1
3	0.913087418	0.934815563	0.995452249	1
4	0.908034361	0.921172309	0.99646286	1
5	0.930773118	0.931278423	0.9646286	1

Table 6 shows the testing results of different classification techniques when using modified GWO for feature selection. Over a set of 5 different experiments, SGD obtained a maximum classification accuracy of 93.24% and an average classification accuracy of 81.99%. SVM obtained a maximum classification accuracy of 94.51% and an average classification accuracy of 83.07%. MLP obtained a maximum classification accuracy of 99.3% and an average classification accuracy of 96.96%. DT obtained a maximum accuracy of 99.8% and an average accuracy of 99.56%.

Table 6. Modified GWO testing results with different machine learning techniques

Experiment	*SGD*	*SVM*	*MLP*	*DT*
1	0.864838394	0.666993144	0.993144	0.997062
2	0.470127326	0.670910872	0.903036	0.997062
3	0.92360431	0.945151812	0.990206	0.998041
4	0.908912831	0.931439765	0.991185	0.993144
5	0.932419197	0.93927522	0.970617	0.993144

The highest Accuracy achieved is 99.8 by GWO. This was achieved at a reduced feature count of 40 out of 75. The lowest feature count was produced by GWO which was 38, this produced an accuracy of 99.7. The average classification accuracy of modified GWO with DT over several experiments is 99.5%. The best classification algorithm is Random Forest.

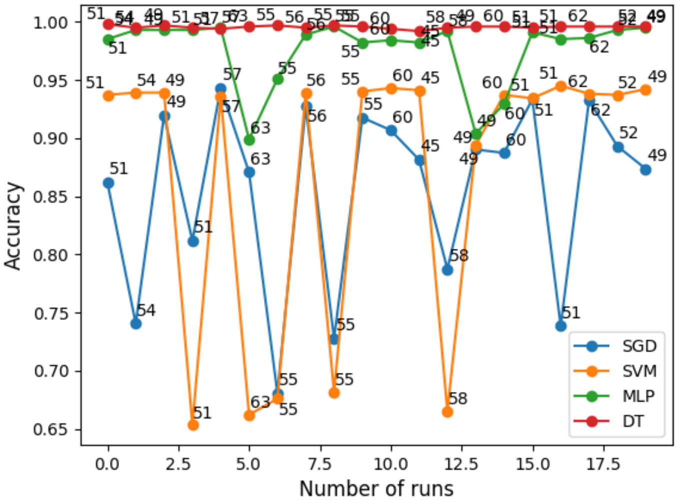

Fig. 4. Comparative analysis of the proposed model with modified GWO

Fig. 4 shows the comparative analysis of proposed model, modified GWO with DT, with respect to the other classifiers. Namely SGD, SVM, MLP. The graph shows that at every iteration, out of 20, the DT algorithm produced best accuracy.

Table 7. Comparative analysis

Optimization Algorithms	*SGD*	*SVM*	*MLP*	*DT*
MVO	0.929480901	0.922624878	0.975514	0.996082
PSO	0.826640548	0.941234084	0.994123	0.997062
Proposed modified GWO	0.92360431	0.945151812	0.990206	0.998041

The table 7 shows the comparative analysis of the proposed model with existing techniques. The MVO optimization produced an accuracy of 0.929480901, 0.922624878, 0.975514 and 0.996082 with classifiers SGD, SVM, MLP and DT respectively. PSO optimization produced an accuracy of 0.826640548, 0.941234084, 0.994123 and 0.997062 for SGD, SVM, MLP and DT respectively. The proposed model obtained the highest accuracy of 99.8%.

4. Conclusion

IDS are the most crucial security mechanisms against the complex and expanding network threats. Performance evolutions for anomaly-based intrusion detection techniques are inconsistent and inaccurate since there aren't enough trustworthy test and validation datasets. In this paper, a modified GWO based feature selection is used along with RF for attack detection. The features that produce the highest accuracy are chosen using a modified GWO method. Stochastic Gradient Descent (SGD) is used in place of the traditional GWO algorithm's fitness function, to modify the GWO and perform feature selection. The RF classifier is trained to recognize attacks using the chosen subset of features. The CICIDS2017 dataset is used to test the proposed model, and its effectiveness has been assessed by comparing it to other machine learning methods that are already in use. The highest Accuracy was attained by GWO, at 99.8. Only 40 out of the 75 characteristics were used to achieve this. When GWO offered 38 features, which was the fewest number available, accuracy was 99.7. Modified GWO with DT obtained an average classification accuracy of 99.5 percent throughout numerous experiments.

References

[1] M. R. Ayyagari, N. Kesswani, M. Kumar and K. Kumar, “Intrusion detection techniques in network environment: a systematic review,” *Wireless Networks*, vol. 27, pp. 1269-1285, 2021.

[2] M. Ozkan-Okay, R. Samet, Ö. Aslan and D. Gupta, “A comprehensive systematic literature review on intrusion detection systems,” *IEEE Access*, vol. 9, pp. 157727-157760, 2021.

[3] N. Dutta, N. Jadav, S. Tanwar, H. K. D. Sarma and E. Pricop, “Intrusion detection systems fundamentals,” *Cyber Security: Issues and Current Trends*, pp. 101-127, 2022.

[4] A. Yazdinejadna, R. M. Parizi, A. Dehghantanha and M. S. Khan, “A kangaroo-based intrusion detection system on software-defined networks,” *Computer Networks*, vol. 184, pp. 1-29, 2021.

[5] A. S. Dina and D. Manivannan, “Intrusion detection based on machine learning techniques in computer networks,” *Internet of Things*, vol. 16, 2021.

[6] Y. Uhm and W. Pak, “Service-aware two-level partitioning for machine learning-based network intrusion detection with high performance and high scalability,” *IEEE Access*, vol. 9, pp. 6608-6622, 2021.

[7] Y. Zhou, G. Cheng, S. Jiang and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Computer networks*, vol. 174, pp. 1-21, 2020.

[8] D. Kapil, N. Mehra, A. Gupta, S. Maurya and A. Sharma, “Network security: threat model, attacks, and IDS using machine learning,” *International conference on artificial intelligence and smart systems (ICAIS)*, March 2021, pp. 203-208.

[9] S. Mirjalili, S. M. Mirjalili and A. Lewis, “Grey wolf optimizer,” *Advances in engineering software*, vol. 69, pp. 46-61, 2014.

[10] L. Bottou, “Stochastic gradient descent tricks,” *Neural networks: Tricks of the trade, Springer, Berlin, Heidelberg*, 2012, pp. 421-436.

[11] M. Yaseen, H. S. Salih, M. Aljanabi, A. H. Ali and S. A. Abed, “Improving Process Efficiency in Iraqi universities: a proposed management information system,” *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 1, pp. 211-219, 2023.

[12] M. Aljanabi and S. Y. Mohammed, “Metaverse: open possibilities,” *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 3, pp. 79-86, 2023.

[13] A. S. Shaker, O. F. Youssif, M. Aljanabi, Z. Abbood and M.S. Mahdi, “SEEK Mobility Adaptive Protocol Destination Seeker Media Access Control Protocol for Mobile WSNs,” *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 1, pp. 130-145, 2023.

[14] H. S. Salih, M. Ghazi and M. Aljanabi, “Implementing an Automated Inventory Management System for Small and Medium-sized Enterprises,” *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 2, pp. 238-244, 2023.

[15] G. Subburayalu, H. Duraivelu, A. P. Raveendran, R. Arunachalam, D. Kongara and C. Thangavel, “Cluster based malicious node detection system for mobile ad-hoc network using ANFIS classifier,” *Journal of Applied Security Research*, vol. 18, no. 3, pp. 402-420, 2023.

[16] G. Perumal, G. Subburayalu, Q. Abbas, S. M. Naqi and I. Qureshi, “VBQ-Net: A Novel Vectorization-Based Boost Quantized Network Model for Maximizing the Security Level of IoT System to Prevent Intrusions,” *Systems*, vol. 11, no. 8, pp. 1-25, 2023.

[17] P. Satyanarayana, G. Diwakar, B. V. Subbayamma, N. P. S. Kumar, M. Arun and S. Gopalakrishnan, “Comparative analysis of new meta-heuristic-variants for privacy preservation in wireless mobile adhoc networks for IoT applications,” *Computer Communications*, vol. 198, pp.262-281, 2023.