International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid Ant Colony Optimization and Deep Learning for Anomaly Intrusion Detection

E. Sandhya[1], R. Benschwartz[2], T. Sathiya[3], M. Sangeetha[4], K. Sreeramamurthy[5], M. Preetha[6]

Submitted: 09/01/2024 **Revised**: 15/02/2024 **Accepted**: 23/02/2024

Abstract: Network security is now a major issue for every distributed system. A lot of risks are becoming harder to identify using antivirus and firewall software. Intrusion detection systems (IDSs) are used to find abnormalities in network traffic to increase security. Security-focused networks are known to be mostly consisting of intrusion detection systems. Determining whether incoming network traffic is abnormal or legitimate is the problem of network anomaly detection. The accuracy of intrusion detection statistics and false alarms of network intrusions resulting from a large amount of network data are among the issues with these kinds of security systems. In IDS, achieving high detection accuracy while minimizing training time is a major problem. However, they are managed inefficiently by the conventional IDS that are currently in effect. To solve these issues with network security, this study proposes IDS that depends on deep learning and hybrid optimization. The NSL-KDD dataset's optimal feature number is selected by the algorithm. Furthermore, this proposed system has combined feature selection with deep learning by using the Ant Colony Optimisation (ACO) algorithm to optimize factors for efficient dataset classification. The proposed approach performance has been evaluated on the current intrusion dataset as NSLKDD. Experimental results indicate that the proposed approach performs better than also achieves good accuracy in comparison to the other recent strategies in NSLKDD.

Keywords: *anomaly Intrusion Detection; Deep Learning; Optimization; Ant Colony; NSLKDD*

1. Introduction

Network security has emerged as a critical challenge for any distributed system [1]. By identifying instances of unauthorized entry, intrusion detection systems are enabled to provide a higher level of security to these networks. Network security architecture often involves intrusion detection systems (IDSs). By identifying anomaly- and signature-based intrusions, they secure the networks. Intrusion detection systems can be categorized into three categories: specification, anomaly, and misuse-based, depending on how attacks are detected [2]. Since IDS databases contain archives of signatures, detection based on misuse can identify established attack behaviors. Due to their signatures have been eliminated from the IDS database, the system's major issue which depends on pattern matching is that it is unable to identify unknown intrusions [3]. IDS detection based on anomaly is predicated on the concept that an attack method can produce behavior that differentiates from a user's typical behavior. It has the potential to help the IDS identify new threats by examining the attack pattern. With its ability to identify any suspicious activity that deviates from regular system operations, anomaly-based detection emerged as the solution to the issue. Among the useful techniques by which an algorithm can be trained to make decisions without explicit programming and with inadequate support from humans is the use of machine learning methods.

Since anomaly-based detection may recognize any abnormal behavior that deviates from the system's normal processes, it was established to address this problem. One of these prospective methods is the use of machine learning algorithms, which allow the system that was developed to develop its ability to make evaluations with the least amount of human involvement and without explicit programming. Supervised, unsupervised, and hybrid learning techniques are the three categories into which machine learning techniques [4]. Numerous security tools are used to address this issue, including an intrusion prevention system (IPS) which analyses discussions in real

[1]*Assistant Professor, Department of CSE (AI), Madanapalle Institute of Technology & Science, Andhra Pradesh, India.*
Email: sandhyaethala@gmail.com
ORCID: 0000-0002-9364-2461
[2]*Associate Professor, Department of Electronics and Communication Engineering, Mar Ephraem College of Engineering and Technology, Kanyakumari.*
Email: rbenschwartz87@gmail.com
ORCID: 0000-0002-2710-2504
[3]*Assistant Professor (Sr.G), Department of CSE, Sona College of Technology, Salem.*
Email: sathiya.t@sonatech.ac.in
ORCID: 0000-0003-0033-1515
[4]*Associate Professor, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai 600123, India.*
Email: sangeethameckanzi2224@gmail.com
ORCID: 0000-0002-2204-072X
[5]*Professor, Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Bowrampet, Hyderabad-500043, Telangana, India.*
Email: sreeram1203@gmail.com
[6]*Professor & Head, Department of Computer Science and Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai.*
Email: smpreetha14@gmail.com, preetha.m.cse@psvpec.in
ORCID: 0000-0001-8483-9871

time to identify attacks, an intrusion detection system (IDS) which gives non-real-time detection of multiple breaches, and a firewall that enables basic session control via a five tuple-based strategy.

IDS collect information from network traffic or system traffic and term it audit data. In the instance that an attack on security is found, a security break has been classified once this audit data has been examined for any violations of the system security policy. The goal of an anomaly detection technique is to identify abnormal network behavior. Both approaches offer the benefits and drawbacks of the respective methods. While the technique used for detecting misuse has a low False Positive Rate, it is unable to identify new attacks. While it may be challenging to generate rules, emerging hazards are recognized by the anomaly detection technique [5].

In this paper, an approach to combine optimization and Deep Learning techniques for the development of a hybrid IDS is proposed. With this hybrid approach, from the enormous dataset, appropriate characteristics are chosen using Ant Colony Optimisation (ACO). The hybrid approach is built by combining a novel deep Extreme Learning Machine (ELM) with an ensemble technique. False Positive Rate (FPR) is reduced and IDS rule development is aided by this hybrid technique. A comparison with other current models is made about their efficiency.

2. Related Works

Studies involving intelligent intrusion detection systems (IDSs) using machine learning techniques for network traffic classification and intrusion detection are abundant in the corresponding research of recent times. Requirements in computing power have traditionally made it difficult to execute full-fledged intrusion detection systems (IDSs) on network gateways, implementing this highly challenging technical challenge. Feature selection is a strategy that is typically used before machine learning algorithms to increase training efficiency. It eliminates the majority of asymmetric information flow and helps distinguish between normal and abnormal factor concerns. This section examines the work being done in the field of intrusion detection systems using machine learning and swarm intelligence methods [5].

The detection of anomalies had been proposed in [6], using Evolutionary Neural Networks (ENN) for classification and a modified Cuckoo Search Algorithm (CSA) called Mutation Cuckoo Fuzzy (MCF) for feature selection. To further thoroughly analyze the search space for potential candidates to allow escape from the local minima, the suggested search algorithm makes use of mutation. Furthermore, the significance of the solution is evaluated as inconsistent with the objective function and the Fuzzy C Means (FCM) clustering approach, which are used to create a fuzzy membership search domain that comprises all conceivable compromise solutions and to get the maximum results for overlapping datasets. The suggested framework is being used accurately for intrusion detection problems and was confirmed using the NSL-KDD dataset. According to experimental results, IDS performance and efficiency can be increased while execution times are shortened by utilizing and selecting the most pertinent features.

The new hybrid intrusion detection system (IDS) was presented in [7] and uses the Ensemble of Feature Selection (EFS) and Adaptive Grasshopper Optimisation Algorithm (AGOA) in conjunction to identify and classify anomalies that arise from computer network attacks. Initially, the filter-based supervised feature selection technique is referred to as ensemble FS, which combines feature rankings produced by different filters to help remove features that aren't important. However, when used to difficult problems, GOA appears to rapidly merge and is easily stranded in local optima. To solve the issue, they have developed an adaptive behavior called GOA called AGOA, which can help forecast network traffic behavior with accuracy. Furthermore, the AGOA method is used to choose precise SVM parameters that do not cause over-fitting. The low false alarm and high detection rate of the suggested model have been proven.

The wrapper feature selection approach for the intrusion detection system is given in [8]. The algorithm manages the technique chosen by the use of a dynamic differential annealed optimizer (DDAO). The suggested algorithm has been standardized for 51 test functions. The dynamic differential annealed optimizer method correlates with the number of quotes optimization strategies. The suggested approach has proven superior to existing approaches; several highly effective situations are demonstrated. The practical engineering optimization of guarded path planning and spring designs is chosen as a problem. DDAO emerged as a global challenge solver with the lowest efficiency, having identified the optimal solution to the spring design challenge as seen by many processes.

Using hybridization between the Artificial Bee Colony (ABC) and Dragonfly (DA) algorithms, [9] provides an innovative binary framework for intrusion detection categorization that increases the accuracy rate of classifying malicious and non-malicious network traffic by training artificial neural networks (ANNs). Initially, the hybrid (ABC) and (DA) models are used to select appropriate weights and biases. These optimal parameters are then used to reconfigure the neural network's intrusion detection approach, enabling it to identify new types of attacks. The performances of the suggested framework and the other eight meta-heuristic methods were evaluated after

they were modified for neural network training. To evaluate the suggested system against others, four different types of intrusion detection analysis datasets were used. As compared to other categorization techniques, the trial findings have demonstrated a significant advancement in ineffective network intrusion detection.

To find the unique poor qualities, the algorithm of opposition self-adaptive grasshopper optimization was used [10] to develop an intelligent method of intrusion detection based on perceptive and mutation approaches. In the suggested approach, reinforcement learning is used in the support vector machine as a gain actor-critic, and the machine's fitness function determines the low false positive and negative rates and high detection rates. The gradient-based method is used in an unpredictable circumstance to aid the agent in rapidly acquiring more precise action selection policies. To demonstrate the effectiveness of the proposed approach, simulation results corresponding to the rate of detection and the rate of false positives are being computed on three different datasets, namely CIC-IDS 2017, AWID, and NSL-KDD. In NSL-KDD, the suggested framework has demonstrated a high detection rate, accuracy, and low false-positive rate.

Utilizing real-time and high accuracy speed, [11] implements the feature selection technique for network intrusion data to detect intrusions on traffic of the real-time network. They use the Cuckoo Search (CS) feature selection technique on three intrusion datasets: Botnet ISCX 2017, NSL-KDD, and KDD Cup 99. The two other evolutionary algorithms, Particle Swarm Optimisation (PSO) and Genetic Algorithm (GA) are compared with the performance of the Cuckoo Search (CS) algorithm. Experiments show that the CS method reduces the number of attributes from 41 to 9 in the NSL-KDD dataset. When it comes to NSL-KDD intrusion datasets, PSO outperforms CS and GA in terms of performance classification.

3. Proposed System

To propose a hybrid anomaly-based intrusion detection system that provides a higher detection rate than the prior work, the proposed approach combines optimization and deep learning methods. This section discusses the general framework of the proposed framework and describes the deep learning and feature selection methods. Classification, feature selection, and preprocessing are the three primary phases of the proposed methodology. This category's classification was utilized in this research to establish a suggested methodology. As previously stated, the classification problem of detection involves the use of predictive algorithms to identify future risks.

Data mining is primarily used for classification purposes. Predicting a sample's class label based on attributes that provide explanations is its primary objective [12, 13]. However, one of the main issues with classification algorithms is an uneven data set [14]. An imbalanced dataset makes data-mining algorithms function inadequately. Therefore, to solve this issue, an appropriate framework is needed. There are also a lot of highly complex features and redundant or ineffective aspects. The substantial search space in these situations could result in a reduction in classification performance. To avoid these unnecessary redundant in this instance without lowering IDS performance, feature selection techniques could be applied. Finding characteristics linked with the learner's classifier facilitates classification, shortens processing times, and enhances accuracy and classification performance. Furthermore, it takes an extended amount of time to classify the massive IDS collection. A dataset that is large in number and features will require an enormous amount of memory and resources to operate. Because IDS datasets typically contain a large number of features and samples, feature selection is therefore very significant.

Preprocessing stage: In this section, the NSL KDD dataset is preprocessed by performing the discretization technique and converting symbolic valued attributes to numeric values.

Phase of feature selection: In this phase, features are selected using Information Gain.

3.1. Feature selection

One of the most important factors in selecting the best NIDS applications is the selection and elimination of redundant features which aid in differentiating between legitimate and fraudulent instances and increase the general efficacy of NIDSs. The selection aims to boost anomaly IDS precision, decrease information redundancies, decrease processing costs of anomaly IDS, and improve the evaluation of conventional network data. The most common approach in this evaluation is enhanced Ant Colony Optimisation (ACO), which evaluates the sensitivity of specific functions. The specified technique for categorizing the abnormal behaviors of DNS and HTTP is most appropriate for the minimum N bug-based criteria [15].

To distinguish between fraudulent instances and those that are legitimate, the approach known as ACO is used to choose pertinent features and eliminate redundant features. The actual foraging behavior of ants provides the framework for ACO [16]. They have the best ability to gather anomalous IDS features and determine the shortest paths from their nests to the anomaly IDS. ACO is a class of algorithms whose solutions are built using information from an issue. The ants currently search randomly for NIDS. When an ant finds a perfect anomaly IDS feature, they integrate it into their colony. A kind of substance known as a pheromone is carried behind them when they

move on the roads. As a result, shorter paths would result in higher levels of anomalous IDS pheromone detection. Pheromone pathways, Equation (i), are used by ants to make decisions. The precision of the solution detected determines the amount of the ground-based pheromone signature. Pheromones interact to generate a higher density along short paths compared to extended paths, which increases the attraction between many ants. Pheromone channels eventually diminish in number due to the evaporation rate. Figure 1 shows the suggested ACO's flow diagram.

As a result, the evaporation processes result in the least amount of local stalling and offer the highest accuracy. The pheromone ratios have to be modified, nevertheless, after each iteration, [16].

$$A_{xy}^{z} = \begin{cases} \frac{(\mu_{xy})^{\alpha}(\delta_{xy})^{\beta}}{\sum c \epsilon D_x^z (\mu_{xy})^{\alpha}(\delta_{xy})^{\beta}} & q \in D_x^z \end{cases} \quad \text{(i)}$$

$$\delta_{xy} = \begin{cases} cw_{xy} \; is \; lower \; G \; \in similar \; class \\ cw_{xy} \; is \; higher \; G \; \notin similar \; class \end{cases} \quad \text{(ii)}$$

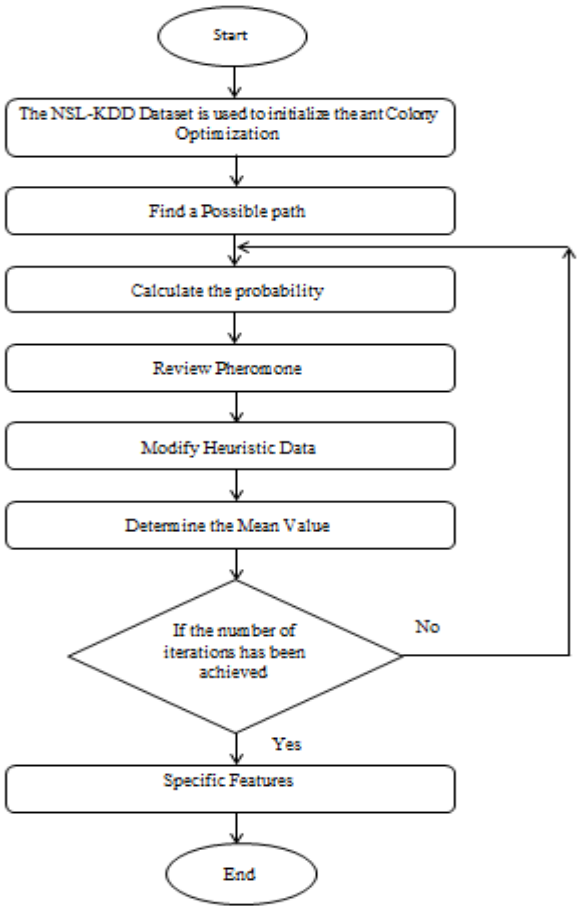

Fig. 1. Enhanced Ant Colony Optimization Flowchart

Where A_{xy}^{z} is the probability that ant k was selected to switch from node x to node y to find the best characteristics for the anomaly IDS network using Eq. (i). The amount of pheromone and experiential information modify this anomaly IDS rate [17, 18]. Although D_x^z is an instance of viable regions that ant z has yet to explore for the optimal selection of features, cw_{xy} represents the distance between the two characteristics that are calculated by Eq. (ii) via the Mahalanobis distance function. μ_{xy} is the heuristic function, δ_{xy} is the value of pheromone at the edge of x and y by Eq. (i), and the values of α and β in Eq. (i) identify the corresponding value of pheromone intensity and heuristic specifics. The distance that exists between two issues in the multivariate space is called the Mahalanobis distance [19]. cw_{xy} is the definition of Mahalanobis distance.

$$cw_{xy} = \sqrt{(x-y)M^{-1}(x-y)^{N}} \quad \text{(iii)}$$

The dataset covariance matrix is represented by SM. Eqs. (iv) and (v) may be implemented for modifying the pheromone in the following manner.

$$\delta_{xy} \longleftarrow \delta_{xy} + \triangle \delta_{xy}^{z} \quad \text{(iv)}$$

$$\triangle \delta_{xy}^{z} = \begin{cases} \frac{F}{g(\psi^z)} w_{xy} \in \psi^z \end{cases} \quad \text{(v)}$$

Equation (vi) provides the evaporation rate.

$$\delta_{xy} \longleftarrow (1-\rho)\,\delta_{xy} \quad \text{(vi)}$$

where F is a constant and ρ is the constant reduction ratio of all pheromones. The accuracy achieved by ant z is denoted by: $g(\psi^z)$. This optimization cycle above reaches 100 after a predetermined number of iterations. Eq. (vi) uses ρ to represent the lowering of the constant factor. To enhance the current situation, people must incorporate the mean function, which has been identified as ρ_new in Eq. (x). (viii) and (ix), which compute the mean values of g1 and g2.

$$\delta_{xy} \longleftarrow (1-\rho_{new})\,\delta_{xy} \quad \text{(vii)}$$

$$\mathrm{C}i_{g1} = \frac{\sum_{x=1}^{H} e_x}{H} \quad \text{(viii)}$$

$$\mathrm{C}i_{g2} = \frac{\sum_{x=2}^{H} f_x}{H} \quad \text{(ix)}$$

$$\rho_{new} = \mathrm{C}i_{g1} * \mathrm{C}i_{g2} \quad \text{(x)}$$

The ρ_{new}, value decreases if the feature standard deviation is less than the ρ_{new} significance, which increases otherwise. Figure 2 displays the suggested IACO flowchart.

An Ant Colony Optimisation (ACO) combines decentralized forecasts, autocatalysis, and competitive objectives to identify the best solution to problems in the context of optimization. Ant behavior is thus imitated in actual life. Numerous optimization issues have arisen since the introduction of the ACO algorithm, such as resource distribution issues, network routing, travel salespeople, and quadratic assignment [20].

The following characteristics of the ants in the ACO technique are:

A. Every ant looks for a feasible partial approach that costs the fewest resources.

B. An ant with the number z may keep data related to its previous path in its memory, N^z. It is achievable to construct effective approaches, analyze existing options, and navigate backward using the information that has been saved.

C. It is feasible to provide a start state ($I^z i$) and several termination conditions (t^z) to an ant (z).

D. Ants begin at a start condition and work their way towards neighboring possible states, partially solving the problem. When at least one of the termination conditions (t^z) for ant z exists, the process comes to an end.

E. An ant z in node p can use probabilistic selection criteria to transfer to node q, which is selected in a suitable neighborhood, $W^z x$. This could be expressed like this:

F. Continuing to any node q in its achievable neighborhood can be done for an ant z in state d =<d-1; x>. $W^z x$ can be expressed as follows: $W^z x$ = {q|(q∈ W_x) \(<d,q>∈D)} d∈ D, where D is a collection of all states."

G. The following functions into a probabilistic principle:

i.Pheromone paths and the principles of heuristics

ii.The ant's knowledge from a prior cycle, and

iii.The limitations of the issue.

H. The ant can modify the pheromone pathways μ_{xy} on the edge (p,q) when it moves from node p to neighbor node q.

I. An ant may modify its pheromone paths, follow its previous travel in reverse, and ultimately vanish once it has developed a solution.

3.2. Deep Learning Based Extreme Learning Machine

The DELM structure utilizes a wide variety of activation functions, a wide variety of hidden layers, and hidden neurons frequently to achieve the ideal configuration for optimizing network security. The three levels that constitute the proposed framework are the data-gathering layer, the pre-processing layer, and the evaluation layer. The program structure uses two sub-layers: one for evaluation and one for estimating. For the experiments, real data was gathered from databases or the cloud. After processing, the data was sent to the data-gathering layer as input. Several data cleaning techniques and inspection approaches are used in the preprocessing stage to eliminate anomalies from the actual data. To strengthen network security towards any intrusive or disruptive function, the Deep Extreme Learning Machine (DELM) framework is implemented at the application layer.

Several intrusion detection systems may be integrated with the DELM approach. Maintaining the optimum detection accuracy typically requires a high percentage of system factors. DELM reduces a number of concerns for network developers, including accessibility and network stability [21]. Given that data transmission and retrieval consume about 80% of a network's resources, approaches for feature and function extraction may be implemented to lower consumption and improve the network's lifespan.

Widely used encoding methods have the potential to enhance resource consumption due to their high computational and memory needs. The goal of DELM is to lower the output requirements for network information compression. Network security involves real-time networking techniques related to safety, scheduling, monitoring, grouping networks, information gathering, error detection, and data integrity. DELM proposes a framework that improves network security's capacity to respond to rapidly evolving circumstances.

To predict safety concerns, energy use projections, transportation, traffic control, etc., a variety of disciplines may apply the Deep Extreme Learning Machine (DELM) [22]. Conventional artificial neural network architectures include a range of tasks and significant learning phases that have the potential to overcome the learning framework [23]. Huang et al. [24] define what is meant by an extreme learning machine. It has also used the back-propagation technique in this forecasting framework through its training phase. In back-propagation, where data streams backward through the network, the neural network changes weights to attain high precision at the lowest error. A feed-forward neural network is typically used in extreme learning machine algorithms, meaning that information travels just one path over a series of layers. Weights remain constant during the network's testing phase, which imports the trained structure and provides the anticipated outcomes. When the framework is validated, testing objectives throughout the network may be conducted using the model with training that was finally provided to the cloud for online use. To enhance network security, mean square error, or MSE, is monitored in the evaluation layer.

While continuously arranging the input weights, DELM integrates several common learning strategies, like backpropagation; it modifies the output weights briefly, and incrementally, and leaves the input weights unchanged. Consequently, it offers a quick and thorough learning capacity. The DELM's procedures will operate in this manner. A training dataset of Z samples (i_x, g_x) in which $i_x \in E_v$ and $g_x \in E_u$ are provided, along with several hidden layer feed-forward neural nets and hidden layer neurons. This several hidden layer feedforward neural network's result might be expressed as follows:

$$\sum_{y=1}^{z} \beta_y K\left(R_y i_x + p_y\right), x \in [1, Z] \quad \text{(xi)}$$

Currently, p_y and R_y serve as learning variables, and the β_y nodes output weight y and K: E. The function that activates is E. This is shown at periodic intervals by an adequate synthesis of many buried Layer Feed-forward Neural networks with null error. Given appearance β_y such that R_y and p_y.

$$\sum_{y=1}^{z} \beta_y K\left(R_y i_x + p_y\right) g_x, x \in [1, Z] \quad \text{(xii)}$$

It can be recognized as

$$J_\beta = \text{G}, \quad \text{(xii)}$$

Where,

$$\text{J} = \begin{bmatrix} K(R_1 i_1 + p_1) & \cdots & K(R_m i_1 + p_m) \\ \vdots & & \vdots \\ \vdots & \cdots & \vdots \\ K(R_1 i_m + p_1) & & K(R_m i_m + p_m) \end{bmatrix} \quad \text{(xiv)}$$

and,

$$\beta = (\beta_1^C \ \ldots \ \beta_m^C)^C \ , G = (g_1^C \ \ldots \ g_M^C)^C \quad \text{(xv)}$$

The equation that follows may be used to get the weights of the outcomes if the volume of metrics over the hidden layer neurons has been determined.

$$\beta = J'\text{G} \quad \text{(xi)}$$

Additionally, matrix J inverse is J'. DELM is a highly effective evaluation technique as a result.

4. Experimental Results and Analysis

The evaluations criteria, datasets used, and planned techniques of comparison are all included in the problem's assumptions. The NSL-KDD dataset is used for various kinds of tests in this component, and significant findings can be derived from the produced outputs. The system used for each group of tests had an Intel (R) Core (TM) i7-2500K processor with GB12 RAM. Matlab2018 was also used to create the simulator software, which is installed on a 64-bit version of Windows 10. Based on parameters, one can compare parameters to evaluate the efficiency of the suggested method.

Information about the true and anticipated classification is provided in the confusion matrix. With the help of this matrix data, the efficacy of the framework could be evaluated. The following data confusion matrix, which is shown in Table 1, is the result when there are two kinds of data: normal and abnormal.

Table 1. Confusion Matrix

Detected Phase			
Abnormal	Normal		
False Negative	True Positive	Normal	Original Phase
True Negative	False Positive	Abnormal	

True Negative (TN): The number of anomalous samples that were accurately identified.

False Negative (FN): The quantity of normal samples that were found to be anomalous.

True Positive (TP): The number of typical samples that were accurately detected.

False Positive (FP): Irregular sample frequency that was shown to be typical.

Accuracy (ACC): The Accuracy Correct Classification Rate (ACC) represents the percentage of accurately predicted samples.

$$\text{ACC} = \frac{TN+TP}{FP+FN+TP+TN}$$

Precision (P): Dependability Precision (as a percentage of output): This criterion is explained as follows:

$$\text{Precision} = \frac{TP}{FP+TP} \times 100$$

Recall (R): This shows the percentage of samples that each class's algorithm was effective in detecting.

$$\text{Recall} = \frac{TP}{FP+TP} \times 100$$

To analyze the suggested approach in terms of intrusion detection into computer networks, we used the NSL KDD dataset. The dataset is provided to fix a few intrinsic issues with the KDD'99 dataset. The overview of the dataset description is provided below. Each sample in the NSL-KDD database has access to 41 features. There have been found four different attack classes that must be separated from normal samples: probing, DOS, U2R, and R2L.

U2R Class: Attacks in which the attacker first obtains the system's user ID before using vulnerabilities to their benefit to get the system's root.

R2L Class: In this instance, an attacker signs into the network pretending to be a user by transmitting data packets across a network and obtaining local access.

DOS Class: An attacker uses various system resources, such as memory and computational power, to entirely interfere with system performance in this type of attack. As a result, the system fails to reply to legislative decisions.

PRB Class: An attacker uses this type of attack to scan a network for weaknesses and gather information. After that,

one must be enabled to quickly identify network behaviors by using information about the computers and services within the network.

The performance of the proposed technique under various conditions, including both normal and attack situations utilizing NSL-KDD, is shown as follows: These simulations were run via simulation parameters in MATLAB.

Table 2 illustrates the anomaly network's performance results. By utilizing the NSL-KDD dataset, IDS developed a methodology. Table 2 demonstrates that the suggested method has been effective in identifying attacks. Significantly, the characteristics of the attacks under discussion exhibit an apparent similarity to typical traffic, making their identification challenging. The proposed method's functioning characteristics are also shown in Table 3.

Table 2. The proposed method's outcomes for identifying attacks

	Normal	**Denial of Service**	**R2L**	**U2R**	**Probe**
Detection Rate (%)	97	98.01	97.24	98.73	98
False Precision Rate	0.24	0.005	0.037	0.019	0.021
Accuracy (%)	97.21	98.37	98	99.12	98.47

Table 3. Results of the suggested technique for identifying Normal and Abnormal attacks

Traffic Type	**No. of Overall Datasets**	**No. of normal Detected Datasets**	**No. of Abnormal detected Datasets**
Normal	154250	124250	3000
Attack	10341	9100	1241
Total	164591	21520	4241

The findings of the suggested technique have been evaluated with those of different approaches.

Table 4. Shows the comparing findings

Classification Techniques	**Accuracy (%)**	**Precision (%)**	**Recall (%)**
KNN	95.18	87.01	91.72
RF	97.24	91.23	95.14
SVM	98.16	89.52	97.34
ACO+DELM	99.87	95.49	98.27

Table 4 and Figure 2 confirm the suggested method performs significantly better in each of the three key parameters. Because it may result in the optimal region, the ACO-DELM technique has proven to choose the best-related characteristics; consequently, it has demonstrated a significantly higher efficiency when compared to other techniques.

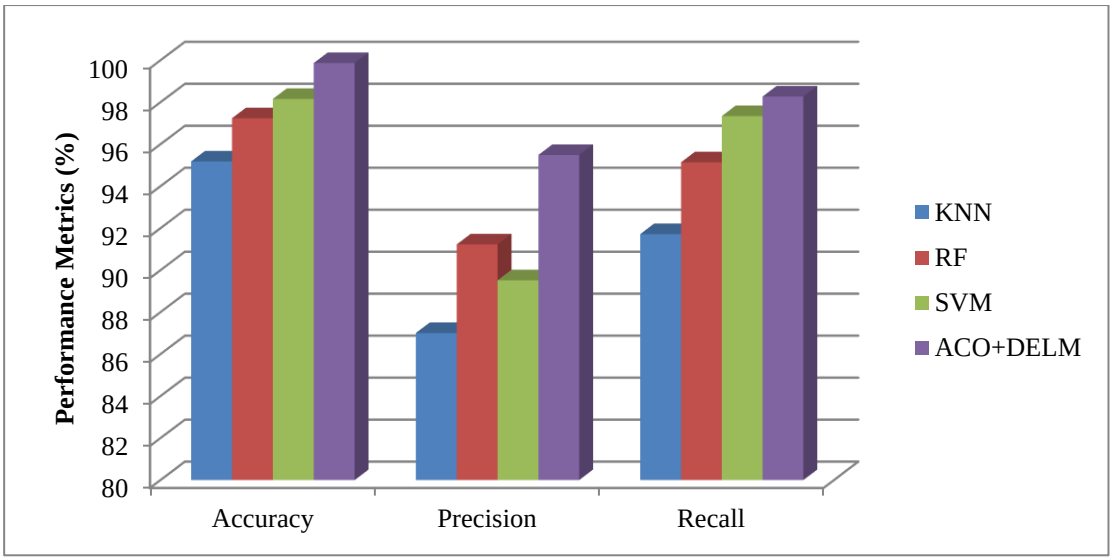

Fig. 2. Performance Analysis of Classification Techniques

The increased number of ant cycles has a beneficial effect on detection optimum because it increases the likelihood of selecting more acceptable features for classification, which is why related features directly improve classification accuracy. However, as AdaBoost is designated to reduce classification errors, if the number of repetitions is controlled at a high level, it will still produce a good result even at a low cycle. Performance when combined provided

that parameters are appropriately controlled. The suggested anomaly network-based method's compute time acquired by the ACO algorithm demonstrates a higher fitness function than other algorithms. Consequently, the suggested ACO-DELM in the suggested method works well for the anomaly detection issue.

5. Conclusion

This research proposes an accurate technique for an anomaly network-based intrusion detection system. To determine how effectively the suggested framework functioned with the NSL-KDD data sets, simulations were run. The IDS's performance is affected by the size and imbalance of the network traffic datasets. Because of the lack of balance, traditional data-mining techniques are unable to correctly identify the minority class. They aim to improve overall accuracy by disregarding the instance of this class, but it's also crucial to use the right minority class protocol instance. Therefore, in the suggested method, unbalanced data has been processed by the ACO meta-algorithm with the right design. The enormous accuracy of the suggested method for categorizing various attack classes serves as the basis for this statement. However, ACO and DELM represent a valuable meta-algorithm that may be applied to optimization issues related to IDS. The suggested technique has been applied to determine the optimal subset of associated attributes for identifying network connections, owing to these algorithms' exceptional performance. The issue of efficiency is also influenced by the parameters regulation technique. The number of replicates, or searches for solutions in the search space, by the colony is quite important given the large number of documents and their attributes.

References

[1] Chkirbene, Z., Eltanbouly, S., Bashendy, M., AlNaimi, N., & Erbad, A. (2020, February). Hybrid machine learning for network anomaly intrusion detection. In *2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT)* (pp. 163-170). IEEE.

[2] Al-Safi, A. H. S., Hani, Z. I. R., & Zahra, M. M. A. (2021). Using a hybrid algorithm and feature selection for network anomaly intrusion detection. *J Mech Eng Res Dev*, *44*(4), 253-262.

[3] Sarvari, S., Sani, N. F. M., Hanapi, Z. M., & Abdullah, M. T. (2020). An efficient anomaly intrusion detection method with feature selection and evolutionary neural network. *IEEE Access*, *8*, 70651-70663.

[4] Aboueata, N., Alrasbi, S., Erbad, A., Kassler, A., & Bhamare, D. (2019, July). Supervised machine learning techniques for efficient network intrusion detection. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-8). IEEE.

[5] Kumari, A., & Mehta, A. K. (2020, October). A hybrid intrusion detection system based on decision tree and support vector machine. In *2020 IEEE 5th International conference on computing communication and automation (ICCCA)* (pp. 396-400). IEEE.

[6] Sarvari, S., Sani, N. F. M., Hanapi, Z. M., & Abdullah, M. T. (2020). An efficient anomaly intrusion detection method with feature selection and evolutionary neural network. *IEEE Access*, *8*, 70651-70663.

[7] Dwivedi, S., Vardhan, M., Tripathi, S., & Shukla, A. K. (2020). Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection. *Evolutionary Intelligence*, *13*(1), 103-117.

[8] Wilson, A. J., & Giriprasad, S. (2020). A Feature Selection Algorithm for Intrusion Detection System Based On New Meta-Heuristic Optimization. *Journal of Soft Computing and Engineering Applications*, *1*(1).

[9] Ghanem, W. A. H., Jantan, A., Ghaleb, S. A. A., & Nasser, A. B. (2020). An efficient intrusion detection model based on hybridization of artificial bee colony and dragonfly algorithms for training multilayer perceptrons. *IEEE Access*, *8*, 130452-130475.

[10] Shukla, A. K. (2021). Detection of anomaly intrusion utilizing self-adaptive grasshopper optimization algorithm. *Neural Computing and Applications*, *33*(13), 7541-7561.

[11] Syarif, I., Afandi, R. F., & Saputra, F. A. (2020, September). Feature selection algorithm for intrusion detection using cuckoo search algorithm. In *2020 International Electronics Symposium (IES)* (pp. 430-435). IEEE.

[12] Mazini, M., Shirazi, B., & Mahdavi, I. (2019). 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*, *31*(4), 541-553.

[13] Ghanem, W. A. H., & Jantan, A. (2016). Novel Multi-Objective Artificial Bee Colony Optimization for Wrapper Based Feature Selection in Intrusion Detection. *International Journal of Advances in Soft Computing & Its Applications*, *8*(1).

[14] Moayedikia, A., Ong, K. L., Boo, Y. L., Yeoh, W. G., & Jensen, R. (2017). Feature selection for high dimensional imbalanced class data using harmony search. *Engineering Applications of Artificial Intelligence*, *57*, 38-49.

[15] Vanitha, S., & Balasubramanie, P. (2023). Improved Ant Colony Optimization and Machine Learning

Based Ensemble Intrusion Detection Model. *Intelligent Automation & Soft Computing*, *36*(1).

[16] Deng, W., Xu, J., & Zhao, H. (2019). An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. *IEEE access*, *7*, 20281-20292.

[17] Rivera, G., Coello, C. A. C., Cruz-Reyes, L., Fernandez, E. R., Gomez-Santillan, C., & Rangel-Valdez, N. (2022). Preference incorporation into many-objective optimization: an Ant colony algorithm based on interval outranking. *Swarm and Evolutionary Computation*, *69*, 101024.

[18] Ali, M., Deo, R. C., Xiang, Y., Prasad, R., Li, J., Farooque, A., & Yaseen, Z. M. (2022). Coupled online sequential extreme learning machine model with ant colony optimization algorithm for wheat yield prediction. *Scientific Reports*, *12*(1), 5488.

[19] Cabana, E., Lillo, R. E., & Laniado, H. (2021). Multivariate outlier detection based on a robust Mahalanobis distance with shrinkage estimators. *Statistical papers*, *62*, 1583-1609.

[20] Kannan, A. V. (2020). Intrusion Detection in Internet Of Things Using Ant Colony Optimisation. *ICTACT Journal On Data Science And Machine Learning*, *1*(03).

[21] Khan, M. A., Rehman, A., Khan, K. M., Al Ghamdi, M. A., & Almotiri, S. H. (2021). Enhance Intrusion Detection in Computer Networks Based on Deep Extreme Learning Machine. *Computers, Materials & Continua*, *66*(1).

[22] Abbas, S., Khan, M. A., Falcon-Morales, L. E., Rehman, A., Saeed, Y., Zareei, M., & Mohamed, E. M. (2020). Modeling, simulation and optimization of power plant energy sustainability for IoT enabled smart cities empowered with deep extreme learning machine. *IEEE Access*, *8*, 39982-39997.

[23] Rehman, A., Athar, A., Khan, M. A., Abbas, S., Fatima, A., & Saeed, A. (2020). Modelling, simulation, and optimization of diabetes type II prediction using deep extreme learning machine. *Journal of Ambient Intelligence and Smart Environments*, *12*(2), 125-138.

[24] Khan, M. A., Abbas, S., Khan, K. M., Al Ghamdi, M. A., & Rehman, A. (2020). Intelligent forecasting model of COVID-19 novel coronavirus outbreak empowered with deep extreme learning machine. *Computers, Materials & Continua*, *64*(3), 1329-1342.