Journal of Theoretical and Applied Information Technology
15th March 2024. Vol.102. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENSEMBLE MACHINE LEARNING ALGORITHM METHODS FOR DETECTING THE ATTACKS USING INTRUSION DETECTION SYSTEM

NAZREEN BANU. A[1], DR.SKB SANGEETHA[2]
[1]Research Scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani campus, Chennai, TN, India
[2]Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani campus, Chennai, TN, India
[1]nazreenbanu8@gmail.com, [2] skbsangeetha@gmail.com

ABSTRACT

Everyday improvement in distributed computing administrations needs more regard to convey the information with security in light of Interruption happening in a decentralized climate. Cloud security needs headways in the Interruption recognition framework in view of material science and web security. The vast majority of the current IDS checking framework wasn't ready to make the component choice and arrangement really to recognize the interruption. Because of expanding more component dimensionality and non-related highlights drives mistakes to accomplish the presentation. To determine this issue, we propose an Ensemble machine learning method for execution. Hybrid Whale optimization algorithm (WOA) using genetic algorithm and Random Forest Integration is executed. Preprocessing is first done to determine the scaling factor and feature margins. Subsequently, the feature outline is marginalised by estimating the User Behaviour Analysis based on Flow and Time-Based Features. Behavioural Features for Frequency of Protocols is computed to highlight the features based on the variation features. Then, at that point, Hybrid Whale optimization algorithm (WOA) to reduce the number of unrelated highlights, a genetic algorithm is used to choose the highlights. In order to identify the IDS, the selected highlights are finally ready for Random Forest Integration. The proposed framework accomplishes superior execution by distinguishing ensemble methods for IDS to sort the interruption level. Also to listen to the dimensionality idea of future varieties to achieve best location exactness contrasted with different frameworks.

Keywords: *Intrusion Detection System, WOGA, RF, Virtual Machine Monitor, NF, FAR.*

1 INTRODUCTION

Intrusion detection systems changed as computer security threat monitoring and surveillance evolved. It is a proactive technology designed to monitor and defend vital IT infrastructure against unauthorised activity. The use of intrusion detection systems (IDS) has significantly increased due to the large volume of data and increasingly complex system attacks[22]. The traditional IDS technique does not comply with cloud requirements since the majority of business and IT sectors are moving towards decentralised architecture, such as cloud computing[23][24]. As a result, in order to function with cloud networks, the IDS needs to be distributed and able to monitor every node in a computing environment. The cloud computing network's intrusion detection system (IDS) is depicted in Figure 1 given below. Users should not rely solely on cloud providers' security infrastructure. It should be feasible for both the user and the service provider to determine a virtual security tool protection structure. The purpose of internet security is to prevent theft or damage to your computer resources and data. Data security refers to the inability of unauthorized parties to read data that is saved on a computer. People have been unaware of computer and cyber security for the last fifteen years. The computer's execution of malicious code as a result of threats or vulnerabilities in the system or application that cause improper user actions and system failures is a security issue.

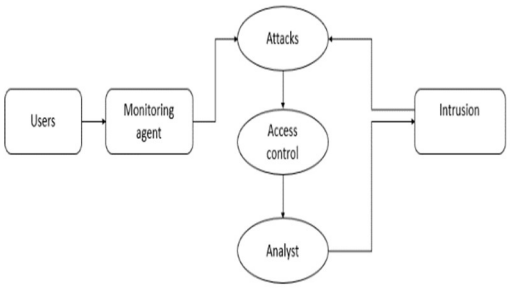

Figure 1: IDS System

Data security refers to the inability of unauthorized parties to read data that is saved on a computer. People have been unaware of computer and cyber security for the last fifteen years. The computer's execution of malicious code as a result of threats or vulnerabilities in the system or application that cause improper user actions and system failures is a security issue. With the computer extended and online, users can browse social media, e-mail, e-commerce, and the Internet. This mainframe is impacted by a multitude of problems, such as malware, spyware, phishing, spam, and scams. Typically, a virus programme replicates itself and damages a computer to cause data loss. Malicious software that tracks your online activities without your knowledge is called spyware. Phishing, scams, malware, and spam all belong to the same category and involve tricking people into clicking on links, images, and other items. The most frequent and enduring security risks end in your inbox. They seem canonical and originate from different sources. Due to their speed, simplicity, and low cost, cyberattacks are typically hard to identify and follow. By enhancing integrity, confidentiality, reliability, and authentication, cyber security helps prevent data loss, computer outages, financial loss, and loss of privacy and personal belongings. In accordance with the security considerations surrounding the transmission of data over networks, various models and planning requirements.

However, a variety of techniques are available in the literature to explain the goal of an intrusion attack, providing access to their host, feature, flow, and other aspects. This flaw lowers the detection rate's performance. The cloud systems rely on a number of variables, including service throughput, resource utilisation, access control, and integrity performance. By increasing the completion of various services, the throughput of the services has been improved; throughput performance has increased with increased service access. When services that have been accessed by legitimate users are successfully completed, the number of services is increased. Thus, performance can be raised and QoS performance can be enhanced by taking these factors into account. Taking the system into account, the hyperplane is able to identify polar routing attacks. Quality of service metrics like packet delivery rate, delay, energy consumption, and packet loss are used to compare these methods. Packet delivery speed is a crucial metric for evaluating the efficacy of signalling protocols in networks. To assess the required protocol performance, several network parameters are taken into consideration. The total number of data packets that are delivered to their destination is known as the packet delivery rate as a percentage of all packets sent from the source.

The authors were inspired to create an effective intrusion detection method after identifying these issues. Malicious activities are problematic because they are among the security risks disseminated by jamming attacks. Although security has been suggested as a solution by some researchers, the issue still exists. There may be a categorization-based attack launch structure. To address this issue, various kinds of network-crashing attacks are possible against each layer. The authors were inspired to create an effective intrusion detection method after identifying these issues. Malicious activities are problematic because they are among the security risks disseminated by jamming attacks. Although security has been suggested as a solution by some researchers, the issue still exists. There may be a categorization-based attack launch structure. To address this issue, various kinds of network-crashing attacks are possible against each layer. By considering all these we propose an ensemble method based

on Hybrid Whale optimization algorithm (WOA) using genetic algorithm and Random Forest Integration is executed. Initially the preprocessing is carried out to find the feature margins and scaling factor.

2 RELATED WORK

The authors were inspired to create an effective intrusion detection method after identifying these issues. Malicious activities are problematic because they are among the security risks disseminated by jamming attacks. Although security has been suggested as a solution by some researchers, the issue still exists. There may be a categorization-based attack launch structure. To address this issue, various kinds of network-crashing attacks are possible against each layer.

A cloud environment primarily needs trust management because a trust relationship is necessary between the user and the cloud service provider. There are inherent risks and challenges associated with the adoption of cloud services [4]. The new approach calls for designing the dependability and integrity of the trust labelling system for communication in order to increase the confidence and trust of cloud service providers.

Data-intensive cloud services are described in [5] as a challenging issue to deliver a personalised and trust-aware service. The quality of service (QoS) of cloud services has been proposed based on trust relationship, trust awareness, and matching method of high-speed resources. Before clustering technique resources were proposed, breaking framework trust aware services were first, and finally trustworthy computing proposed a service resource based on real-time and dynamic monitoring data [6]. The potential for data to leak as a result of an attack by other cloud users presents for the integration of an intrusion detection system with a cloud-based robot system based on integrated learning, new fuzzy semi-supervised learning techniques are effective [7]. More precisely, the fuzzy-based approach will be used for data that is not labelled and the integrated learning method will be used for training after the data has been labelled.

Since machine learning technology has entered the IDS mainstream, handling the same functions is a crucial issue. Two modes, the Intrusion Detection System (IDS) and the Classification System, have been proposed to TIDCS and TIDCS-Accelerated to facilitate a secure network. These modes are based on the Whale Optimisation Algorithm (WOA) [8]. TIDCS advises routine system cleaning in order to assess the trustworthiness of the participating nodes.

The Virtual Machine Introspection (VMI)-based design's adaptive and effective security architecture offers fine-grained virtual machine monitoring [9]. It makes use of the Virtual Machine Monitor (VMM) layer on VM Guard, an operating system-independent platform. Application software is used to record the program's execution by employing breakpoint injection techniques.

Even though the network traffic data process is becoming more and more important, intrusion detection system (IDS) treatments are still ineffective. New data has been suggested for the network anomaly detection process after processing the model [10][21]. The model suggests that it is possible to optimize the number of functions (NF), accuracy rate, recalls, and false alarm rate (FAR).

One of the main issues that is receiving a lot of attention lately is cloud security. Unfortunately, it is very difficult to detect the attack and explore the zombie because of the cloud computing system [11]. The cooperative anomaly identification system to identify both external and insider attacks from the cloud's core. In order to provide an overview of cooperative Network Intrusion Detection Systems (NIDS), a demonstration of how to resolve block chain issues pertaining to trust management and data privacy has been provided.

Resolving cloud-based security threats is challenging for traditional security systems. The new architecture of NIDS for thorough analysis is called the Coordinated and then NIDS Framework; it is built on solutions in an SDN-based environment and will be carried out through a variety of experiments [12].

The cloud computing model of the cloud environment involves issues with security and different types of network attacks. Automatic Encoder (AE) uses a new architecture that combines feature extraction with an advanced neural network (ANN) classification. To alter the test's technical design parameters, AE and DNN have been optimised [13]. K-Dependent Bayesian Network Improvement (KDBN) reduces complexity and enhances the structure model that explains the dependencies of relationships between the variables in systems. In the end, it builds IDCM using the MAP standard, which was introduced to achieve a high efficiency in network intrusion detection. Cloud computing-related technologies were receiving a lot of attention. Despite all of the cloud's benefits, malicious services and attacks of all kinds can be hosted by attackers [14]. Put your trust in modern security techniques to safeguard cloud tenant transactions. Our idea leverages the advantages of trusted computing technology to enhance the development and implementation of security policies and procedures in a cloud environment.

Traditionally, algorithms that rely on average consensus have been vulnerable to attacks using data injection. In order to localise and detect nodes using a Neural Network (NN) model, three node methods for recognising issues have been proposed [15]. These methods describe behavioural-based intrusion detection systems (IDS) that use the behavioural hit feature algorithm to monitor intrusion activity. Deep neural evolutionary networks are utilized to extract both hidden and deep fault signatures from alarm data that is gathered from cloud data centre connections in large quantities. The accuracy of obstacle location is increased by a significant global search function with tilt-free control [17]. Using warning sets and fault propagation models as inputs, a fault localization technique based on Deep Neural Evolutionary Networks (FL-DNEN) is utilized to identify suspect fault regions.

In a heterogeneous cloud edge environment, resource sharing via SDN and NFV is employed across private and public networks [18]. A trustworthy platform needed a reliable way to develop a trust management system, and blockchain technology—which leverages transparency and decentralization—provides that trusted service management [19]. Outlined how users could easily share and work together on files using the Cloud storage platforms [20]. Nevertheless, each file needed an owner who had the authority to decide on access control unilaterally [21]. The idea of shared ownership has nothing to do with the current cloud. Cooperation is severely restricted when an owner has the ability to remove a file or revoke access without first consulting other collaborators [22].

Multilayer perceptron (MLP) and graph convolutional networks (GCN) are combined in the novel approach known as MLP_GCN. Additionally, a technique to handle the approximate cycle time related to the point cloud body while tracking the data in the training process is suggested in this framework. DL is layered with ML systems. In the ML model, Physically Unclonable Functions (PUFs) are suggested to assist the attacker in breaking hardware authority. Deep Neural Networks (DNNs) can be readily broken, even with small inputs, as publications in the field have demonstrated. The following are the limitations: An ideal recognition system is often achieved by employing fewer samples with a higher number of features to validate the proposed solution by characterizing different types of malicious behaviours exhibited by users. Detecting a specific attack type will be a future extension for this research. There is no need for a foreign key as required in encryption methods. The system ensures behavioral patterns which include abnormal and regular

events. ML models should primarily be used as a defensive rather than an offensive tactic. It detects only the known attacks with less classification rate in Network Layer. Smaller numbers of keys are required for a large network compared to symmetric keys. The classified algorithms such as decision tree, random forest and so on will be more in depth on the classified detection effect of cloud computing IDS. This model will provide a secure environment in cloud computing and protect the important resources and data on the Host Machine. It produces data clustering and removes outliers of malicious samples and splits them into several small clusters for the preparation of GAN training. It was shown that our IDS can detect replay and drop attacks very efficiently. For tempering attacks, the detection ratio can be improved by exploiting the relationship between the signals. It is also found that scenarios with balanced and larger records result in better performance.

3. PROBLEM STATEMENT

Existing research work addressed only the common attacks on cloud computing by employing ML algorithms. Computational time is very high and provides less accuracy rate. There is an increase in the high false alarm rate. The Intrusion detection system has security issues in cloud services.

4. DATASET

In this section, the description of the dataset is done. The dataset taken is the Kdd1 dataset that is taken from UCI Machine Learning Repository. The details about the dataset are given. Kdd1 Dataset. In the research work related to Cloud based Intrusion Detection System, Kdd1 Dataset has been commonly used and studied on by many researchers. The dataset is available in the UAV Intrusion Detection System. There are 41 attributes in the dataset. There are approximately 4.9 million instances or records. A sample of the dataset is given below in table 1. The dataset is widely used in research and benchmarking for intrusion detection, but it's important to acknowledge the field of machine learning and intrusion detection.

5. THEORETICAL CONCEPTS AND ALGORITHMS

In order to advance IDS development, a double layer IDS utilising a Hybrid Whale optimization algorithm (WOA) using genetic algorithm and Random Forest integration method is put into practise. Preprocessing is first done to determine the scaling factor and feature margins. Subsequently, the feature outline is marginalised by estimating the User Behaviour Analysis based on Flow and Time-Based Features. Behavioural Features for Frequency of Protocols is computed to highlight the features based on the variation features. Then, it is practical to identify the defect features in order to reduce the non-related features using the HWO using genetic algorithms.

In order to identify the IDS, the chosen features are lastly trained using a Random Forest integration approach (RFI). The suggested Architecture HWOA is depicted below in Figure 2. The suggested system classifies the intrusion level by identifying double layer IDS, which yields high performance. In order to achieve the best detection accuracy in comparison to the other systems, it is also important to reduce the dimensionality of future variations. The dataset is run through an outlier preprocessing model before being handled in order to confirm the existence of the data, remove noise, and scale the margin between ideal and actual verification in order to eliminate unwanted data and facilitate effective feature selection.

5.1 User Behaviour Analysis based on Flow and Time-Based Features

The tolerance affected feature limits are used to estimate the delay performance of effective feature

communication at this stage. The attribute marks advancement and an indicator of the likelihood that data reading distance x is the true value of the actual and ideal margins is D(x), a discriminant score. ex: the actual data event's expected value. For a given noise, the

generator's output power is represented by G (z). Discriminant estimates of the likelihood of a false event are represented by z. z)). The expected value (the expected value of G(z) for all false events that are actually generated) for all random inputs to the generator is denoted by the letter Ez. It displays the application of the Minimax principle algorithm.

Attribute	Description
duration	Length of time (in seconds) the connection was active
protocol_type	Type of the protocol used in the connection (e.g., tcp, udp, icmp)
service	Network service on the destination (e.g., http, smtp, ftp)
flag	Status of the connection: normal or error status
src_bytes	Number of data bytes transferred from the source to the destination
dst_bytes	Number of data bytes transferred from the destination to the source
land	Indicates whether the connection is from/to the same host/port; binary (1 if true, 0 if false)
wrong_fragment	Number of "wrong" fragments in the connection
urgent	Number of urgent packets in the connection
hot	Number of "hot" indicators in the connection
num_failed_logins	Number of failed login attempts
logged_in	Indicates whether the user is logged in; binary (1 if true, 0 if false)
num_compromised	Number of compromised conditions
root_shell	Indicates whether root shell is obtained; binary (1 if true, 0 if false)
su_attempted	Indicates whether "su root" command attempted; binary (1 if true, 0 if false).
num_root	Number of "root" accesses.
num_file_creations	Number of file creation operations
num_shells	Number of shell prompts
num_access_files	Number of operations on access control files.
num_outbound_cmds	Number of outbound commands in an ftp session
is_host_login	Indicates whether the login belongs to the "hot" list; binary (1 if true, 0 if false).
is_guest_login	Indicates whether the login is a "guest" login; binary (1 if true, 0 if false)
Count	Number of connections to the same host as the current connection in the past 2 seconds
srv_count	Number of connections to the same service as the current connection in the past 2 seconds.
serror_rate	Percentage of connections that have "SYN" errors
srv_serror_rate	Percentage of connections to the same service with "SYN" errors
rerror_rate	Percentage of connections that have "REJ" errors
srv_rerror_rate	Percentage of connections with "REJ" faults to the same service.
diff_srv_rate	Proportion of connections to various services
srv_host_diff_rate	Percentage of connections made for the same service to several hosts
dst_host_serror_rate	Proportion of connections with "SYN" faults to the destination host.
dst_host_srv_serror_rate	Proportion of connections with "SYN" problems to the destination host and service.
dst_host_rerror_rate	Proportion of connections with "REJ" faults to the destination host.
dst_host_srv_rerror_rate	Proportion of connections with "REJ" faults the destination host and service.

Table 1: Kdd1 Dataset Attribute Description

Pseudo code

set the pre-processed features to initial

Determine the appropriate and real margin for defect labels.

Perform m steps for every training iteration.

Node a abnormal feature limits {q_1,...,q_a} and

transforms with mean abute rate.

Normal features of Node a: {q_1,…..q_a} from actual data

To attain to update the actual feature limits

$$\Omega_{\infty_i} \frac{1}{a} \sum_{x=1}^{a} \left[log\ log\ X\left(q^{(x)}\right) + log\ log\ (1-X)\left(A(p^{(x)})\right)\right] \quad (1)$$

End for

Features m noise features and{p_1,……..p_a} transforms with generator

Retain the actual feature limits

$$\Omega_{\infty_r} \frac{1}{a} \sum_{x=1}^{a} log\ log\ (1-X)(A(p^{(x)})) \quad (2)$$

End for

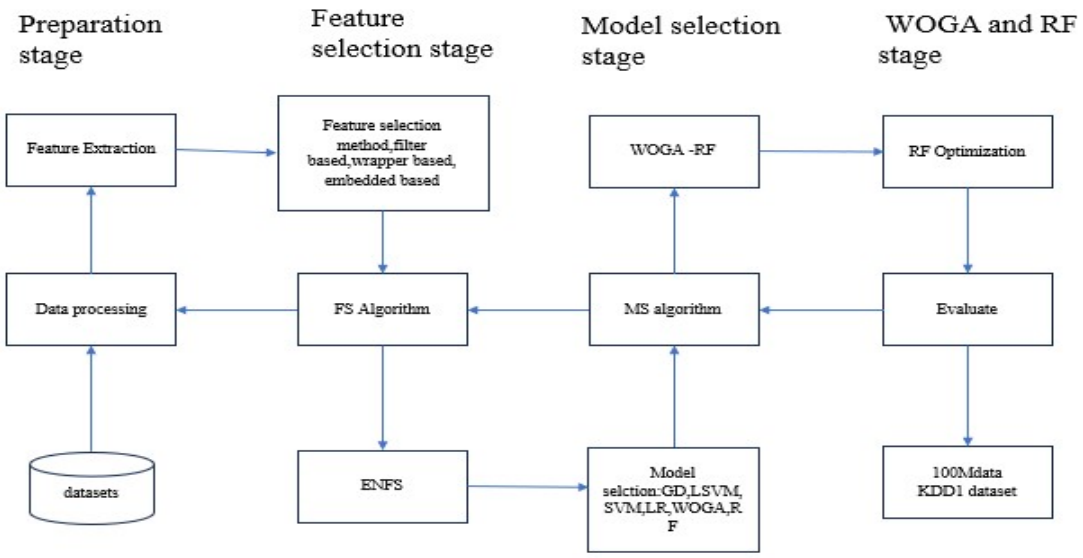

Figure 2: Proposed Architecture

The cross entropy between the generated and true distributions yields this expression. Reducing log(1−X(A(p)) is the same as minimising generator loss. This is due to the fact that while the generator is operating, log(X(q)) has no impact on the terms. The following function is maximised by the discriminator and minimised by the generator in a GAN loss function. Take into consideration a flaw in the data transmission and malicious activity detection features. The following describes how the suggested method operates: Sniffer labels are used to predict non-variation features in a network with a uniform distribution, which is used to estimate the support rate.

5.2 Behavioural Features for Frequency of Protocols

At this point, the cumulative data transmission rate for behavioural intrusions is carried out in a manner akin to the user dependencies on network transmission. This considers the data transmission rate and the user's behavioural exploration factor when accessing the data in the network nodes. The following parameters are also used in addition to the ones used in the transmission rate evolution.

o Abnormal behaviour indicates occupancy and utilisation ceilings

o Rate of feature channel switching

The following equation is used to apply parameter exploration once the parameters have been determined.

$$h(q) = \left[u^{-\sum_{x=1}^{X} \left(\frac{q_x}{\alpha}\right)^{2a}} - 2u.\prod_{x=1}^{X} cos^2(q_x)\right] \quad (3)$$

$$\alpha - exploration\ factor, a \\ - sample\ count, X \\ - parameter\ count, \\ q - input\ value$$

Next, using the following equation, the output is given as the Stretched V cos wave function.

$$T(q) = \sum_{x=1}^{X-1} (hp_{x+}^2 + hq_x^2)^{0.25} \left[cos^2\left\{50\left(hp_{x+1}^2 + hq_x^2\right)^{0.1}\right\} + 0.1\right] \quad (4)$$

The logarithmic distribution is computed for every case, and the stratified output's superscript symbols are defined. The calculation of coefficients using the circularly symmetric complex Gaussian criteria, which accounts for the communication's defect IDS channel interference. After that, the values that have been investigated are used to inform learning decisions that identify intrusion factors.

The main purpose of feature selection is to relate from the optimized state. Measuring the amount of predictive bit information surrounding the class—if any—in the presence of feature filtration and the associated class distribution is the only information available. Following observation of the E value, the information gain (HI) E of the class attribute of the designated attribute E lowers the uncertainty of the value y. The Algorithm model is depicted below in Figure 3.

$$A = F|E \quad (5)$$

The uncertainty value of subset features F is estimated using the entropy value (W), where E and F are random features (s_1….s_l and q_1….q_l) from the dataset.

$$W(F) = I\sum_{x=1}^{m} J(s_x)log_2(J(s_x)) \quad (6)$$

where J(s_x) represents the prospects that came before for all values of F. The provisional entropy of F given that E is substituted assumes the uncertainty surrounding the value of F after detecting standards of E.

$$W(F|E) - \sum_{b=1}^{c} J(q_x)\sum_{x=1}^{m} J(s_x|q_b)log_2(J(s_x|q_x)) \quad (7)$$

If V(F | E) > V(F | I), then the attribute q is deemed higher than the attribute I if the switching level F attributes this scale. It enables the selection of important attributes according to the information gained determined by weighing the significance of each attribute class.

5.3 Hybrid Whale optimization algorithm (WOA) using genetic algorithm

This step involves the implementation of the Hybrid Whale Optimisation Algorithm (WOA), which uses a genetic algorithm to select features based on a multi-level Fuzzification feature defect offloading system. This reduces the execution time and dimension factor. The recommended task is further improved using the Hybrid Whale optimisation algorithm (WOA) to reach the performance rate), which uses a genetic algorithm for feature optimisation. Eqn 1 illustrates how the proposed strategy can offload the tasks generated

maximum feature dependencies MD x={1,2,3…,N} to the central cloud.

$$X_{(c,d)}\{min \quad if\ feature\ e_{d,x}\ is\ max\ in\ RX\ \ 1 \quad if\ feature\ e_{d,x}\ is\ ideal\ max\ if\ task\ e_{d,x}\ is\ executed\ in\ core\ cloud \quad (8)$$

where c = [1, 2, 3,..] Which feature data is the mean? The number of features in the Kth iteration is denoted by RX, d={t_(1,) t_(2,) t_3,…t_c}. The following describes the HWOA processing state: The offloading decision of 0 in the above equation indicates that the feature is unrelated in and of itself, and that otherwise it is at maximum scaling based on class dependencies.

The representation of t_1 and t_2 conditions is the process of this fuzzified feature relation.

First condition: feature limits are scaled from MD himself and will be real if capacity(p_1<ρ_1^sbs).

Second condition : Some features are unrelated if capacity(p)_1>ρ_1^sbs and there is a relative feature margin.

Third condition : If there are features that are near the maximum range and capacity (p_1>ρ_1^sbs).

In order to take into consideration the total number of features carried, K={1,2,3,…N}. Additionally, c={1,2,3,..n} is based on relative features, and d={1,2,3,…M} is the count of features from actual limits. Process limit at feature computations: for every MD user i, where i∈N at similar iteration t, features are generated as c_i^t. When it comes to defect features from illness 1, the limits of the representation feature are,

$$ZRX_{(c,d)} = \theta_e \lambda(c,d) = \theta_k s(c,d) \quad (9)$$

$$KRX_{(c,d)} = \frac{j(c,d)}{h(x)}$$

where j(c,d) is the total size of the nth feature at kth features are actual, θ_l is the feature support of the local RX, θ is the total feature ideal, and h_l is the feature dispensation degree. The total relation feature e in the disorder 2 scenario is stated in the equation below. Additionally, the equation below defines the total feature that was chosen.

$$f^e = \{M^e_c\}_{c\in K} \quad (10)$$

$$E^e = \{E^e_q\}_{x\in c}$$

The difference between two activation limits is used to access the ideal and actual limits, which represent the membership function.

$$€ = \{\alpha, \beta, \gamma, \delta, \theta\}, € \in E^e \quad (11)$$

Where γ is the estimated feature limits margin in the range of 20-100, β-difference of 0-1, β-number of feature inputs with 0-50, δ-margin between 2-40 and θ-variation. The following equation declares the membership function and its fuzzy set, respectively.

$$\psi_A(E): E \to [0,1] \forall x \in E \quad (12)$$

$$B = \{(E, \psi_A(E))\}: x \in E \quad (13)$$

Using the gaussian principle support factor, the triangular membership function is declared based on the upper and lower margins.

$$\psi_B^{triangular}(Z) = \{0 \quad if E \le t\ \frac{a-t}{b-t} \quad if\ t < E < h\ \frac{p-x}{q-x} \quad if\ c < E < g\ 1 \quad if X \ge g \quad (14)$$

The feature difference is represented by r_0 and r_1, and the differential feature limits are denoted as d_(t_u) and d_(t_d), respectively.

$$ZRZH^j_{e_{(n,x)}} = \frac{d_{t_u}}{r_i} \forall x \in E\ and\ \forall i \in [0,N] \quad (15)$$

$$ZRZH^r_{t_{(m,i)}} = \frac{d_{t_d}}{r_i} \forall x \in X\ and\ \forall i \in [0,N] \quad (16)$$

$$KRZH_{t(m,i)} = ZRZH^j_{e_{(n,x)}} + ZRZH^r_{t_{(m,i)}} \quad (17)$$

The relational feature limits are estimated based on the predominant feature dependencies.

$$ZH_{(c,d)} = \sigma j(c,d) \quad (18)$$

$$KH_{(c,d)} = \frac{j(c,d)}{u_c} + \frac{\lambda}{f_c} \quad (19)$$

Thus, σ stands for actual feature limit, b for MD fitness, and f_c for feature limits derived from variation. Each workflow population's fitness function is based on the features. In GWO, the socialisation and hunting behaviours of the group of wolves are mathematically modelled to carry out the optimisation process using,

$$X = |H \times E_J - E_{(K)}$$

$$E = |F_{P(t)} - H \times X| \quad (20)$$

Here, H_X and E_((t)) represent the prey's locations. Iteration (t) the coefficient vectors C and D are computed as

$$H = |2 \times a \times r1 - a|$$

$$Z = 2 \times r2 \quad (21)$$

Whereas r1, r2-feature support margins [0,1], and a-linearly decreased 2 to 0 over the iteration, as measured by C-[-a, a] and E- [0,2].

$$b = 2 - t \times \frac{2}{No.of\ iterations} \quad (22)$$

The active position of can be altered by adjusting the parameters C and E in accordance with the best feature limits. The finest keys are framed as (X_1, X_2, and X_3). The alpha (α), beta (β), and delta (δ) are responsible for placing the prey.

$$X_\alpha = |Z_1 \times E_\alpha - E|$$

$$X_\beta = |Z_2 \times E_\beta - E|$$

$$X_\delta = |Z_3 \times E_\delta - E| \quad ,(23)$$

And

$$E_1 = E_\alpha - H_1 \times W$$

$$E_2 = E_\alpha - H_2 \times W_\alpha$$

$$E_3 = E_\alpha - H_3 \times W_\alpha \quad (24)$$

The next iteration population is calculated as

$$E_{(k+1)} = \frac{E_1 + E_2 + E_3}{3} \quad (25)$$

Based on |Z|<1 and the fitness position of feature margins, the location of the Wolves is updated. To marginalise the features, the α, β, and δ wolves update the wolves' strike units. To divert the wolves from pursuing the target, C_1 arbitrarily takes values above or below -1. Grey Wolf's fitness function represents the challenge of offloading to reduce execution time and energy consumption. It is the total weighted amount of the mobile device's execution time and energy usage during the workflow. The total execution duration is regarded as

$$KLW = wk \times (ZKL_{(c,d)} + ZKZ_{(c,d)} + ZKH_{(c,d)} \quad (26)$$

The feature variation limits of mth features of nth device executed in RX, edge cloudlet, and central cloud are denoted by ZKT, ZKZ, and ZKH. The energy consumption of the mth task on the nth device in the RX, edge cloudlet, and central cloud is represented by the variables ZHJ, ZHZ, and ZHH.

$$WZH_{(c,d)} = (1 - xu) \times (ZHM_{(c,d)} + ZHM_{(c,d)} + ZHH_{(c,d)}) \quad (27)$$

Wt is the RX-based weight vector, ranging from 0 to 1. A lower value of wt indicates that the RX battery needs to be conserved, and a higher value of wt increases the execution time and energy. Therefore, the declared weight is 0.6.

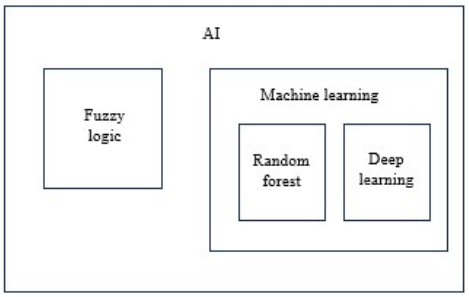

Figure 3: Algorithm model

***Pseudocode*:**

Hybrid Whale Optimization (WOA) with Genetic Algorithm (GA) for Uncommon Attack Detection

Define the problem-specific parameters

Set WOA parameters

max_iterations_woa = 100

population_size_woa = 20

a = 2 # Parameter for updating positions in WOA

Set GA parameters

max_iterations_ga = 50

population_size_ga = 10

crossover_rate = 0.8

mutation_rate = 0.1

Initialize the WOA population

woa_population = initialize_woa_population(population_size_woa)

Main optimization loop

for iteration in range(max_iterations_woa):

Update positions of whales using WOA

update_positions_woa(woa_population, a)

Evaluate fitness of whales

evaluate_fitness(woa_population)

Sort whales based on fitness

sort_whales(woa_population)

Select the best whale

best_whale = woa_population[0]

Apply GA operators on the best whale

For ga_iteration in range(max_iterations_ga):

Select parents using tournament selection

parents = tournament_selection(woa_population, population_size_ga)

Apply crossover and mutation

offspring = crossover_and_mutation(parents, crossover_rate, mutation_rate)

Evaluate fitness of offspring

evaluate_fitness(offspring)

Replace the worst individual in the WOA population with the best individual from offspring

replace_weakest_whale(woa_population, offspring)

Identify uncommon attacks using the best whale

detect_uncommon_attacks(best_whale)

Output the best whale (solution)

best_solution = woa_population[0]

print("Best Solution:", best_solution)

Detect uncommon attacks using the best solution

detect_uncommon_attacks(best_solution)

To implement these functions based on the details of your problem, and the parameters such as population size, mutation rate, and crossover rate may need to be adjusted based on your specific requirements and

experimentation. This reduces the feature dimensionality based on spectral ratio to compare with support vectors in the Neural network. The classifier predicts the result based on the class by IDS margins.

5.4 Random Forest Algorithm

A technique known as ensemble learning, Random Forest builds a large number of decision trees during training and outputs the mean prediction for regression issues or the mode of the classes for classification problems based on the individual trees. Uncommon attacks refer to malicious activities that are not well-represented in the training data. Traditional rule-based IDS systems may struggle with detecting such attacks, making machine learning models, like Random Forests, a valuable addition.

The architecture of a Random Forest integration method from figure 4 which is given below is defined by the combination of decision trees and given below, the strategy of random feature selection, and the ensemble learning framework. Its strength lies in the ability to create a diverse set of trees that collectively provide robust and accurate predictions From the figure 5 given below the integration of Random Forest into network security involves applying the machine learning algorithm to analyse and classify network flow data for the purpose of detecting anomalous or malicious behaviour. This process enhances the ability of security systems to identify and respond to potential threats in real-time.

The classification of data using Intrusion Detection Systems (IDS) from figure 6 given below involves the process of categorizing network or system activities as either normal or potentially malicious based on predefined patterns or models. The classification process is a fundamental component of intrusion detection, enabling the automated identification and response to potentially malicious activities within a network or system. The effectiveness of the IDS depends on the quality of training data, the chosen algorithm, and continuous refinement based on evolving threat landscapes.

The formula for simple logistic regression is x*v + b = y. The input data is represented by the instance, the coefficient or weight that takes its place, b is a representation of the bias and the result or predicted about the information is represented by y. The most important thing is how this framework flows and transforms data.

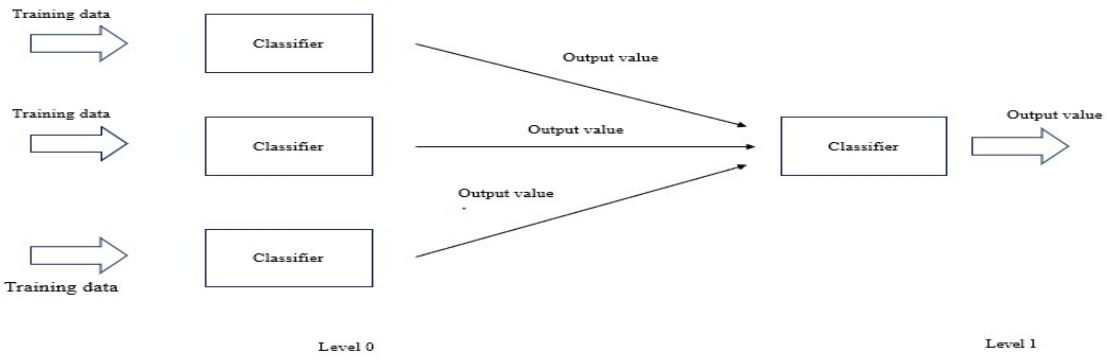

Figure 4: Random Forest Algorithm

Journal of Theoretical and Applied Information Technology
15th March 2024. Vol.102. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

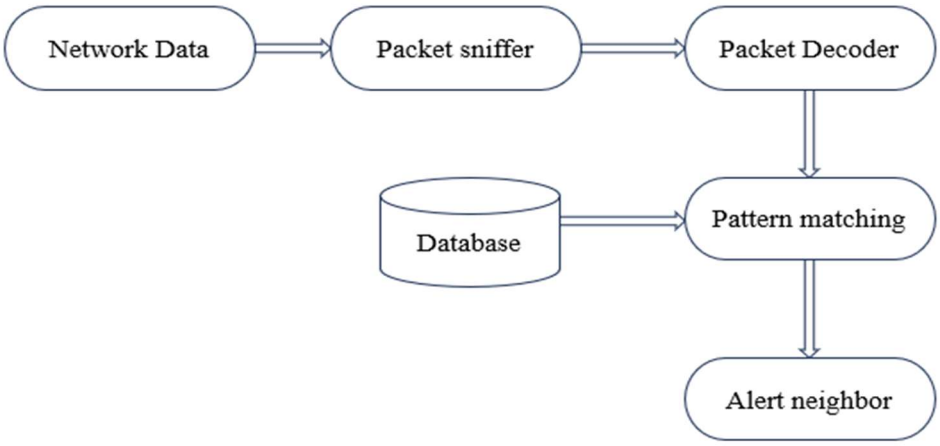

Figure 5: Network Flow Of RFI

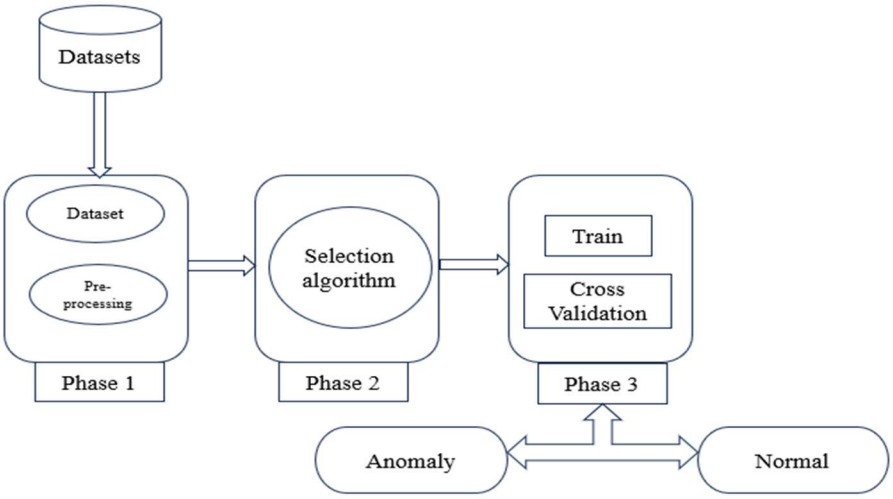

Figure 6: Classification Of Data Using IDS

Pseudocode:

Random Forest Parameters

num_trees = 100

max_depth = None # Maximum depth of individual trees

min_samples_split = 2 # Minimum samples required to split a node

Function to train a single decision tree

def train_decision_tree(data, labels):

tree = DecisionTree() # You'll need to implement a DecisionTree class

tree.train(data, labels, max_depth, min_samples_split)

return tree

Function to implement the Random Forest algorithm

def random_forest_train(data, labels, num_trees):

forest = []

```
for _ in range(num_trees):

# Randomly sample data with replacement (bootstrapping)

bootstrap_data, bootstrap_labels = random_sampling_with_replacement(data, labels)

# Train a decision tree on the bootstrap sample

tree = train_decision_tree(bootstrap_data, bootstrap_labels)

# Add the trained tree to the forest

forest.append(tree)

return forest

# Final prediction is based on majority voting

final_predictions = majority_voting(predictions)

return final_predictions
```

Finding Accuracy

```
# Assuming you have a set of test data (test_data) and corresponding labels (test_labels)

test_predictions = random_forest_predict(forest, test_data)

# Function to calculate accuracy

def calculate_accuracy(predictions, true_labels)

correct_predictions = 0

total_instances = len(predictions)

for i in range(total_instances):

if predictions[i] == true_labels[i]:

correct_predictions += 1

accuracy = correct_predictions / total_instances

return accuracy

# Calculate accuracy

accuracy = calculate_accuracy(test_predictions, test_labels)

# Print the accuracy

print("Accuracy:", accuracy)
```

In this pseudocode, the train_ decision_ tree function trains a single decision tree, the random_ forest_ train function trains the Random Forest by creating an ensemble of decision trees, and the random_ forest_ predict function makes predictions using the ensemble. The calculate_accuracy function calculates the accuracy by comparing the model's predictions with the true labels. The actual implementation may also include additional considerations such as handling categorical features, pruning, and optimizing for performance.

6.RESULTS AND DISCUSSION

An implementation is tested that has been suggested with publicly available cloud darknet datasets in an Anaconda environment using Python. IDS can be successfully identified using comparison metrics using a confusion matrix, such as time complicity, false ratio, sensitivity, specificity, and accuracy of categorization. This volume offers a descriptive presentation of the findings and discussions from the recommended methods. Additionally, thirty services are used in this, with the parameters as indicated in Table 1.

Based on the Multi-Factor Intrusion Detection System, the comparison algorithms are MLP-GCN and support vector machine (SVM). The confusion matrix computes the following parameters. Table 2 shows the IDS accuracy performance versus the number of services using various techniques, including the suggested IDS-based feature Analysis Model (HWOA-RFI) and SVM, LSVM, and MLP-GCN. Figure 7 describes below the accuracy of the comparison method using the dark net dataset and evaluating the results based on the TP, TN, FN and FP ranges. Compared to the most advanced model, the suggested strategy increased accuracy by 10%, indicating that the proposed model is more accurate in predicting

threats than the current models. The impact of intrusion detection system performance on categorization accuracy with different services, such as 20, 40, and 60 is shown in Figure 8 given below. For 60 services, the suggested HWOA-RFI method achieved 98.3%. Additionally, the prior SVM scored 78.3%, WOA scored 81.8%, and MLP-GCN scored 87.8%. However, in comparison to other approaches, the suggested method yields superior performance.

Figure 9 below shows the proposed method achieved precision by 0.94, recall by 0.89, F1 by 0.92%. In contrast to earlier techniques, the state of art method performed better in malware prediction in the network flow. The precision and F1- score is increased by 4% than the existing state of art model. The impact of the suggested sensitivity performance in comparison to earlier methods is shown in Table 3. The sensitivity performance for IDS detection using the HWOA-RFI algorithm is displayed in Figure 10 below. The suggested algorithm yields 96% sensitivity performance for 60 services; in comparison, the current algorithm yields 76% sensitivity performance for SVM, 81% sensitivity performance for WOA, and 85% sensitivity performance for MLP-GCN for 60 services.

TABLE 2: Computed Values And Surroundings

Keys	Values
Tool for Simulation	Notebook Jupyter and Anaconda
Language used for simulation	Python
The dataset's name	Kdd1-dataset
Lack of morals and environment	700/ 3000
Number of classes	High / medium / low

TABLE 3: Accuracy Of Intrusion Detection In Relation To The #

Intrusion Detection Accuracy in % vs #			
Comparison techniques	20-Features	40- Features	60- Features
SVM	63.7	67.6	69.3
LSVM	71.2	74.1	83.8
MLP-GCN	75.7	82.4	87.5
HWOA-RFI	81.9	93.6	97.8

Table 4: Sensitivity Performance's Impact

Comparison Techniques	20- Features	40- Features	60- Features
SVM (%)	47.9	58.6	62.8
LSVM (%)	59.7	63.2	72.6
MLP-GCN (%)	68.8	71.7	79.5

HWOA-RFI (%)	73.9	79.3	85.5

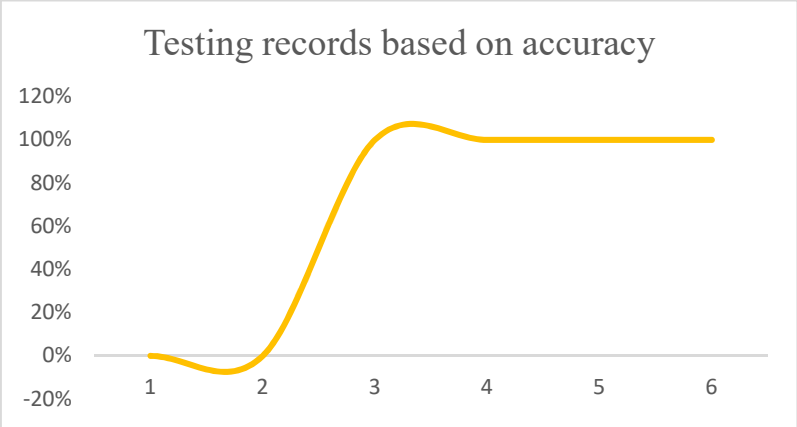

Figure 7: Analysis Of Accuracy

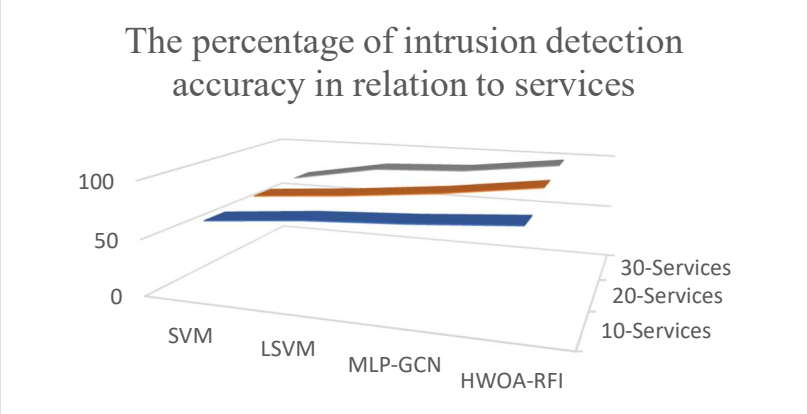

Figure 8: Performance Of IDS Accuracy

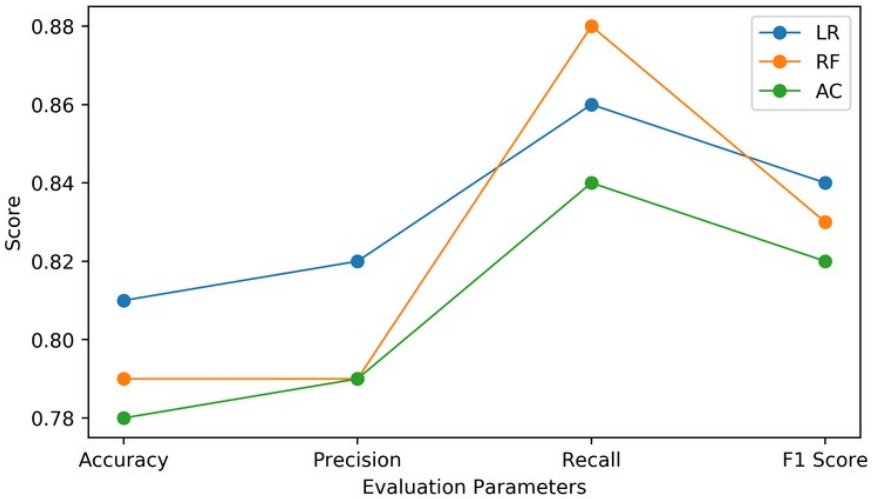

Figure 9: Analysis Of Performance Matrix

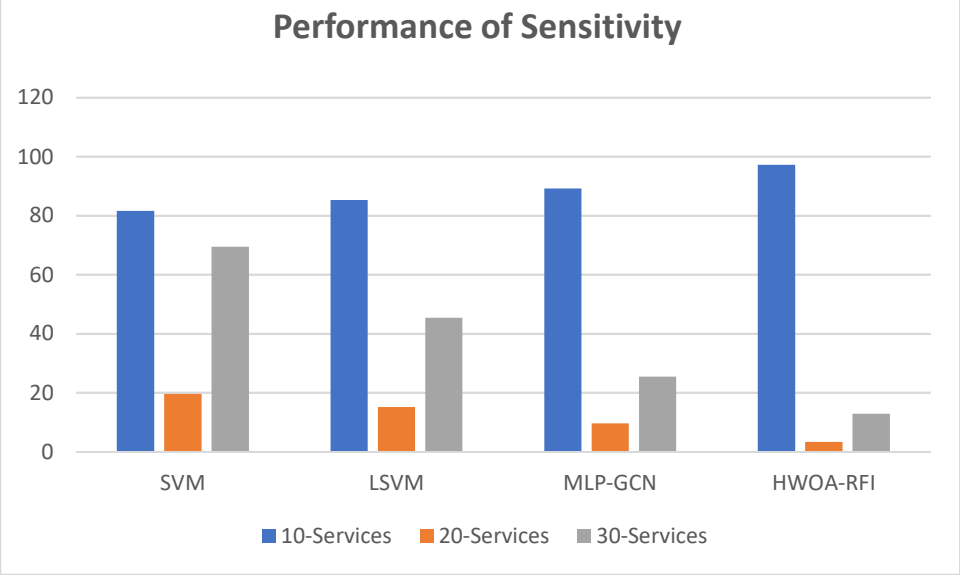

Figure 10: Performance Of Sensitivity

TABLE 5: Impact Of Specificity Performance

Comparison techniques	20- Features	40- Features	60- Features
SVM (%)	63.3	72.4	81.5
LSVM (%)	71.7	79.5	83.2
MLP-GCN (%)	73.7	81.5	89.2
HWOA-RFI (%)	81.3	87.5	95.1

TABLE 6: Examination Of The False Classification Ratio

False Classification Ratio in % vs #			
Comparison techniques	20- Features	40- Features	60- Features
SVM (%)	27.3	23.5	19.5
LSVM (%)	21.6	18.1	13.4
MLP-GCN (%)	19.2	14.2	10.4
HWOA-RFI (%)	10.4	5.6	2.5

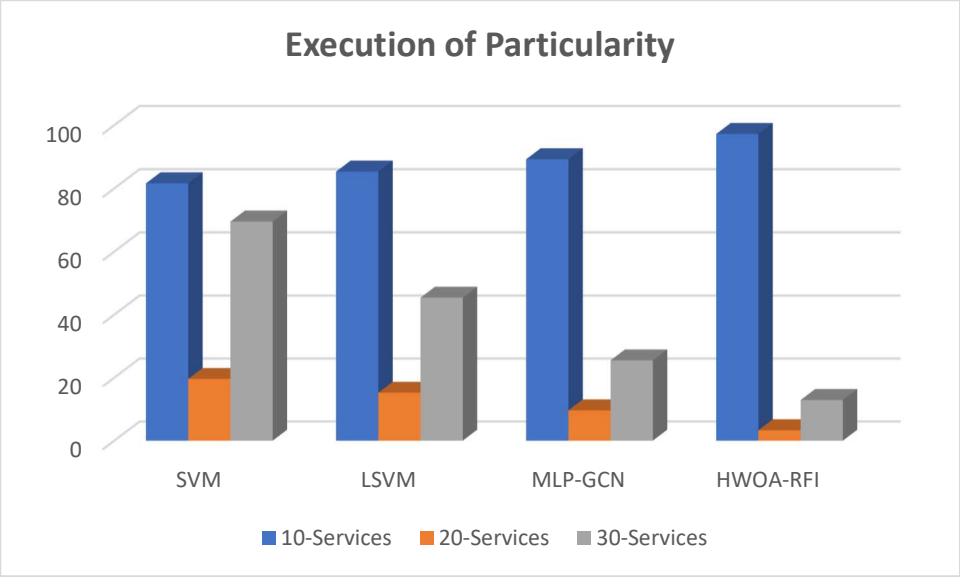

Figure 11: Execution Of Particularity

The analysis of specificity performance measures across various service numbers, including 20, 40 and 60 is presented in Table 4. Compared to earlier methods, the suggested technique yields superior results. Figure 11 below shows the results of a comparison between the suggested and earlier approaches' analyses of specificity performance. Similar to the current algorithm results, which show that WOA has an 82% specificity performance, MLP-GCN has an 86% specificity performance, and SVM has a 77% performance for 60 features, the proposed HWOA-RFI algorithm has 97% specificity performance for 60 features. Examination of the erroneous classification ratio Table 5 lists the suggested comparison's performance with earlier approaches. The performance of the false classification ratio for IDS with 20, 40, and 60 features is shown in Figure 12 below. This graph's X-axis compares different methods, and the Y-axis shows how each method's performance progressively declines. For thirty services, the suggested Service Specific Payload Inference Analysis Model (HWOA-RFI) approach shows a performance in erroneous classification was 1.3% together with SVM's 20.7% false classification performance, the WOA and MLP-GCN methods each achieve 16.9% and 12.6%, respectively.

The effect of performance in terms of time complexity versus number of services is shown in Table 6. In contrast to SVM's 67.2 seconds, WOA's 58.6 seconds, and MLP-GCN's 41.2 seconds, the suggested FGWO- LSVNN has 28.3 seconds for IDS classification. The suggested Service Specific Payload Inference Analysis Model (HWOA-RFI) technique's time complexity performance is shown in Figure 13 along with comparisons to other techniques such as SVM, LSVM, and MLP-GCN. In addition to the Y-axis, which shows the time complexity performance in seconds for each method, the X-axis in figure presents comparison methods. Nevertheless, compared to earlier methods, the suggested method yielded results with less time complexity.

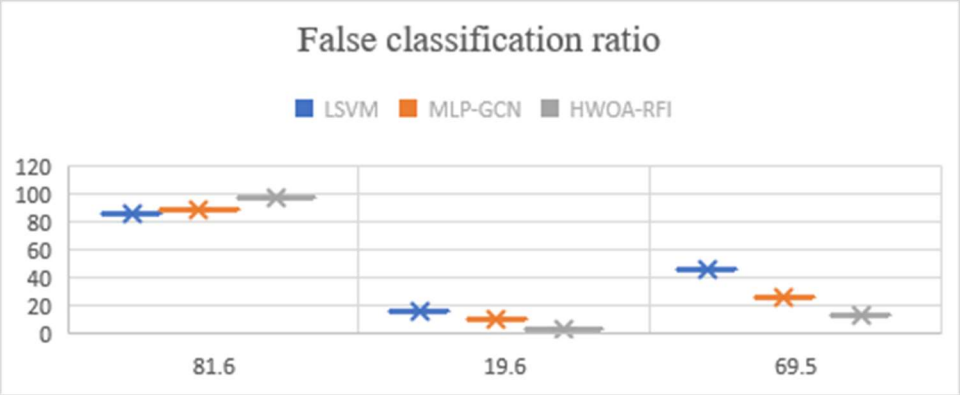

Figure 12: False Classification Ratio

TABLE 7: Efficiency In Terms Of Time Complexity

Number of seconds versus time complexity			
Comparing techniques	20- Features	40- Features	60- Features
SVM	58.1	67.5	71.8
LSVM	37.3	42.4	47.7
MLP-GCN	41.2	49.1	52.5
HWOA-RFI	19.4	25.5	29.4

TABLE 8: Performance Based On A Range Of Metrics

Comparison techniques	% of Detection Rate	% False Ratio	Time Complexity in Seconds
SVM	81.1	19.4	69.4
LSVM	85.3	15.3	45.3
MLP-GCN	89.4	9.7	25.6
HWOA-RFI	97.1	3.5	12.4

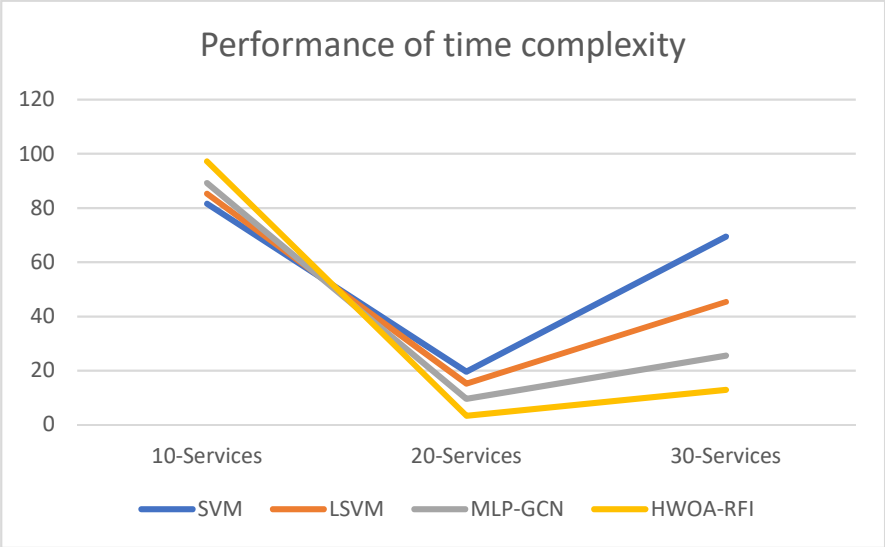

Figure 13: Performance Of Time Complexity

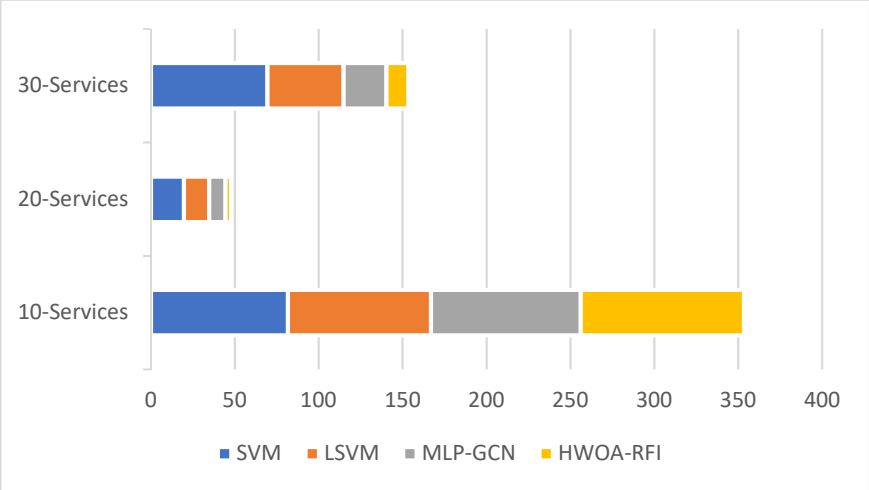

Figure 14: Selection Of Features

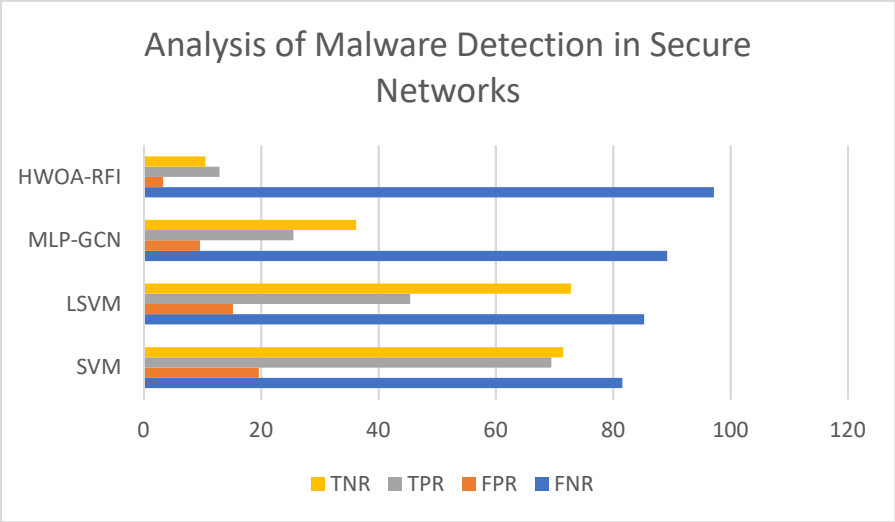

Figure 15: Analysis Of Malware Detection In Secure Networks

Table 7 displays the recommended HWOA-RFI performance of various measurements based on time complexity, false ratio, and detection rate.. The performance of the suggested HWOA-RFI techniques is superior to that of the current approaches. The figure 14 is based on the feature selection using training and testing features (duration, resp_bytes, resp_ip_bytes and proto_tcp, protocol, orig_ip_bytes) from Darknet dataset. And these features identify the risk, low risk and normal level. Figure 15 is described that CPU, False Negative Rate (FNR), False Positive Rate (FPR), True Negative Rate (TNR), and False Discovery Rate (FDR) are effectively resolved to show the classifier performance for misclassification of features.

7.CONCLUSION

The novel service-centric HWOA-RFI 's proposed goal is to detect cloud intrusions. Since different user's access cloud services at different times and with different frequencies, and because these variations occur between users, this phenomenon has been used to analyse communication. The features' outline is marginalised using an estimated User Behaviour Analysis based on Flow and Time-Based Features. Behavioural Features for Frequency of Protocols is computed to highlight the features based on variation features. Then, in order to reduce the non-related features, the features are selected using the Hybrid Whale Optimization using Genetic algorithm (HWOA). In order to identify the IDS, the chosen features are lastly trained using a Random Forest Integration method (RFI). IDS detection is aided by the significance of data analysis based on service access rate and throughput. The IDS classification performance of the suggested HWOA-RFI method is 95.2%, the false rate is 1%, and the time complexity is 16.3 seconds. Compared to other approaches, the suggested method performs better for IDS detection.

DATA AVAILABILITY

The dataset used is taken from UCI Intrusion Detection System and the links are given as follows:
https://archive.ics.uci.edu/dataset/130/kdd+cup+1999+data.

REFERENCES:

[1] Ribeiro, J., Saghezchi, F. B., Mantas, G., Rodriguez, J., & Abd-Alhameed, R. A. (2020). Hidroid: prototyping a behavioral host-based intrusion detection and prevention system for android. *IEEE Access*, *8*, 23154-23168.

[2] P.Parsamehr et al., "A Novel Intrusion Detection and Prevention Scheme for Network Coding-Enabled Mobile Small

Cells," in IEEE Transactions on Computational Social Systems, vol. 6, no. 6, pp. 1467-1477, Dec. 2019, doi: 10.1109/.TCSS2019.2949153

[3] Illy, G. Kaddoum, P. F. de Araujo-Filho, K. Kaur and S. Garg, "A Hybrid Multistage DNN-Based Collaborative IDPS for High-Risk Smart Factory Networks," in IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 4273-4283, Dec. 2022, doi: 10.1109/TNSM.2022.3202801.

[4] M. Zekri, S. E. Kafhali, N. Aboutabit and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, Morocco, 2017, pp. 1-7, doi: 10.1109/CloudTech.2017.8284731

[5] Haghighat, M. H., & Li, J. (2021). Intrusion detection system using voting-based neural network. Tsinghua Science and Technology, 26(4), 484-495.

[6] Ogawa, K., Kanai, K., Nakamura, K., Kanemitsu, H., Katto, J., & Nakazato, H. (2019, March). IoT device virtualization for efficient resource utilization in smart city IoT platform. In 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 419-422). IEEE.

[7] Dutt, I., Borah, S., & Maitra, I. K. (2020). Immune system-based intrusion detection system (IS-IDS): A proposed model. IEEE Access, 8, 34929-34941.

[8] Y. Sun, L. Hou, Z. Lv and D. Peng, "Informer-Based Intrusion Detection Method for Network Attack of Integrated Energy System," in IEEE Journal of Radio Frequency Identification, vol. 6, pp. 748-752, 2022, doi: 10.1109/JRFID.2022.3215599.

[9] J. Wu, Y. Wang, H. Dai, C. Xu and K. B. Kent, "Adaptive Bi-Recommendation and Self-Improving Network for Heterogeneous Domain Adaptation-Assisted IoT Intrusion Detection," in IEEE Internet of Things Journal, vol. 10, no. 15, pp. 13205-13220, 1 Aug.1, 2023, doi: 10.1109/JIOT.2023.3262458.

[10] R. Bitton and A. Shabtai, "A Machine Learning-Based Intrusion Detection System for Securing Remote Desktop Connections to Electronic Flight Bag Servers," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1164-1181, 1 May-June 2021, doi: 10.1109/TDSC.2019.2914035.

[11] Zhou, M., Li, Y., Xie, L., & Nie, W. (2019). Maximum mean discrepancy minimization based transfer learning for indoor WLAN personnel intrusion detection. IEEE Sensors Letters, 3(8), 1-4.

[12] Gao, B., Bu, B., Zhang, W., & Li, X. (2021). An intrusion detection method based on machine learning and state observer for train-ground communication systems. IEEE Transactions on Intelligent Transportation Systems, 23(7), 6608-6620.

[13] De Araujo-Filho, P. F., Pinheiro, A. J., Kaddoum, G., Campelo, D. R., & Soares, F. L. (2021). An efficient intrusion prevention system for CAN: Hindering cyber-attacks with a low-cost platform. IEEE Access, 9, 166855-166869.

[14] Naseri, T. S., & Gharehchopogh, F. S. (2022). A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. Journal of Network and Systems Management, 30(3), 40.

[15] Heidari, Arash, and Mohammad Ali Jabraeil Jamali. "Internet of Things intrusion detection systems: A comprehensive review and future directions." Cluster Computing (2022): 1-28.

[16] Gupta, N., Jindal, V., & Bedi, P. (2022). CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Computers & Security, 112, 102499.

[17] Rawat, S., Srinivasan, A., Ravi, V., & Ghosh, U. (2022). Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network. Internet Technology Letters, 5(1), e232

[18] Asad, H., & Gashi, I. (2022). Dynamical analysis of diversity in rule-based open source network intrusion detection systems. Empirical Software Engineering, 27, 1-30.

[19] Abushark, Yoosef B., A. Irshad Khan, Fawaz Alsolami, Abdulmohsen Almalawi, Md Mottahir Alam, Alka Agrawal, Rajeev Kumar, and Raees Ahmad Khan. "Cyber

security analysis and evaluation for intrusion detection systems." Comput. Mater. Contin 72 (2022): 1765-1783.

[20] Le, Thi-Thu-Huong, Yustus Eko Oktian, and Howon Kim. "XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems." Sustainability 14, no. 14 (2022): 8707.

[21] Khalaf, O. I., Ogudo, K. A., & Sangeetha, S. K. B. (2022). Design of graph-based layered learning-driven model for anomaly detection in distributed cloud IoT network. Mobile Information Systems, 2022, 1-9.

[22] Sangeetha, S. K., Mani, P., Maheshwari, V., Jayagopal, P., Sandeep Kumar, M., & Allayear, S. M. (2022). Design and analysis of multilayered neural network-based intrusion detection system in the internet of things network. Computational Intelligence and Neuroscience, 2022.

[23] Aravindhan, K., Sangeetha, S. K. B., & Kamesh, N. (2022, March). Improving Performance Using Hybrid Framework Iot Communication In Cloud Computing. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 1654-1658). IEEE.

[24] Sangeetha, S. K. B., & Dhaya, R. (2023). An evolutionary predictive security control architecture for healthcare transactions in IoT-based cloud storage. In Unleashing the Potentials of Blockchain Technology for Healthcare Industries (pp. 95-105). Academic Press.