International Journal of Engineering and Management Research
e-ISSN: 2250-0758 | p-ISSN: 2394-6962
https://ijemr.vandanapublications.com

Peer Reviewed & Refereed Journal
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.46

Intelligent Intrusion Detection Categorization using Support Vector and Fuzzy Logic

Monis Tariq[1] and Mohd. Suaib[2]
[1]PG Student, Department of CSE, Integral University, Lucknow, INDIA
[2]Associate Professor, Department of CSE, Integral University Lucknow, INDIA

[1]Corresponding Author: monistariq3@gmail.com

Received: 02-06-2023 Revised: 16-06-2023 Accepted: 30-06-2023

ABSTRACT

Intelligent intrusion detection is a crucial component in ensuring the security of computer networks and systems. Traditional intrusion detection systems (IDS) often struggle to handle the complexities and uncertainties associated with modern network environments. To address these challenges, a novel approach is proposed in this paper that combines fuzzy logic and support vector classification. The main objective of this research is to enhance the accuracy and efficiency of intrusion detection by leveraging the complementary strengths of both fuzzy logic and support vector classification. Fuzzy logic allows for the representation of uncertainty and imprecision in data, making it well-suited for handling the vagueness often present in network traffic and system logs. On the other hand, support vector classification is a powerful machine learning technique known for its ability to handle high-dimensional data and effectively classify complex patterns.

Keyword-- Preprocessing, Feature Extraction, Support Vector Classification

I. INTRODUCTION

Due to its low cost and pay-as-you-go model, cloud computing is a new technology that has been embraced by businesses of all sizes. With its distinctive and widespread capabilities, it has changed the IT industry. Organizations choose the cloud because it eliminates the expensive infrastructure and upkeep requirements. It offers three service models: infrastructure as a service (such as Amazon Web Service, Eucalyptus, and Open Nebula), platform as a service (such as Google App Engine and Microsoft's Azure), and software as a service (such as Google Apps) [1]. Cloud computing can now offer flexibility, usability, scalability, and on-demand network access to a common pool of reconfigurable computer resources thanks to virtualization. The service-oriented architecture of the cloud computing paradigm has significantly changed how services are delivered and managed. Infrastructure, platform, and apps as a service make up its three-tier design, each of which is weak and susceptible to security vulnerabilities. Attackers may potentially jeopardize the resources, data, and virtualized infrastructure of a cloud computing system, which could lead to the emergence of new attack types. When a cloud with enormous storage and computational power is assaulted by attackers already present in the cloud, the issue could get worse and be more serious. Additionally, as a result of their vulnerability at the virtual machine level, the use of hypervisors and virtual machines in the cloud also produces security risks like DDoS attacks. Customers' names, social security numbers, birth dates, and other personally identifiable information were stolen by hackers, who vanished without leaving a trace of their illicit behavior. This was regarded as one of the biggest breaches in history, costing the corporation a loss of $275 million [2,3,4]. In order to safeguard the resources, infrastructure, and sensitive data from hackers, it is essential to develop a system that is considerably more robust. When it comes to protecting client data and resource assets in cloud computing from security risks, intrusion detection systems are crucial. One of the cutting-edge security technologies, it may shield network data from hostile activity. Given that cloud computing is distinct from conventional computer systems, it should be carefully built and compatible with its features. Additionally, it must have effective cloud-specific attack detection capabilities [5,6].

The main objective of Intelligent detect intrusions categorization using support vector and fuzzy logic to enhance the accuracy and efficiency of intrusion detection by leveraging the complementary strengths of both fuzzy logic and support vector classification. Fuzzy logic allows for the representation of uncertainty and imprecision in data, making it well-suited for handling the vagueness often present in network traffic and system logs. On the other hand, support vector classification is a powerful machine learning technique known for its ability to handle high-dimensional data and effectively classify complex patterns [7,8].

II. MODULE FOR FUZZY IMPLEMENTATION

Fuzzy logic compares different metrics of the number of missed packets to different factors. The problem that was not resolved by existing techniques is now solved by fuzzy techniques. By eliminating numerous uncertainties, the fuzzy logic

Peer Reviewed & Refereed Journal
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.46

technique is incredibly simple to use and produces precise results. The number of packets dropped by a particular node that is present near the destination node is greater than T1 and less than T, and the number of packets dropped by that same node that is present near the source node is greater than T3, as a result of the three attacks used in this paper. These three measures are called m1, m2, and m3, and they are used to calculate fuzzy values.

The threshold levels are selected based on how the network behaves and how often assaults occur. Since the black hole attacker node will discard every packet, 20% of all packets are taken as the threshold value in this study. The threshold value is 5% of the total packet because the gray hole attack threshold value is lower than the black hole attack threshold value. P is the universal set of all misbehavior and attacks P= "p1, p2, p3" and M denotes the universal set of all measures M= "m1, m2, m3". L stands for the nodes that make up the topology, which is specified as L=1, 2, 3,..., l, where l stands for the total number of nodes[10, 11].

The relation matrix depicts the L occurrence with regard to measure M and is calculated by L, M, and X. Where Rs (l, m) (l E L, m EM), RS = L X M. The range of m in node l is specified by this matrix. M, X, and P are used to generate the matrix RO, where the matrix denotes an occurrence relation matrix and M denotes the frequency of attacks P. Where Rs(m,p)(m E M, p E P), RO = M X P. M X P is used to generate the matrix RC, which indicates the recurrence of M in relation to the attacks P, Rs(m, p)(m E M, p E P).|

III. FUZZYESTIMATION

Based on the values of the matrices RS, R1, R2, R3, and R4, fuzzy estimate is performed. The values of the matrices are computed based on the threshold values T and T1. Consider the matrices RS, whose value is expressed as RS[node number][member function value] and is derived using the matrices' node number and accompanying member function value. The value of the member function is determined using the trapezoidal membership approach [12,13].

Value of membership = (x-a)/(b-a)

Where x is the threshold value, an is the number of forwarded packets, and b is the number of discarded packets. Depending on the previous value, the R0 and RC are presumptive.

$$R0 = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix} \quad R1 = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Based on membership value, the RS value is determined and updated in the table. The remaining values for R1, R2, R3, and R4 are computed using this RS value and updated in the table [14,15].

I. the indication of occurrence R1 = RS*R0 (the operator * denotes the maximum and minimum composition of RS and R0), II. the indication of conformability R2 = RS * RC, and III. V. Non-symptom indicator R4 = (1-RS) * R0; III. Non-occurrence indication R3 = RS * (1-R0);

Preprocessing: Raw network data and system logs are collected and preprocessed to extract relevant features. The fuzzy logic approach is applied at this stage to handle the imprecise and uncertain nature of the data. In a cloud context, intrusion detection includes gathering and digesting raw network data and system logs to identify pertinent features. Here is a general description of the procedure:

Data gathering: In the cloud environment, many sources of raw network data and system logs are gathered. Network traffic information, server logs, firewall logs, authentication logs, and other pertinent sources are included in this[16,17].

Preparing the data for feature extraction requires preprocessing the collected data. Data cleaning, filtering, normalizing, and consolidation techniques may be used during the preprocessing stage to get rid of noise, unnecessary data, and duplicate entries [18,19].

After the data has been preprocessed, pertinent features must be extracted. The process of feature extraction entails converting raw data into a format that machine learning algorithms may utilize to spot patterns and anomalies. Numerous methods, including statistical analysis, frequency analysis, time-series analysis, and data aggregation, can be used [20].

Feature Extraction and Selection: To reduce dimensionality and increase the effectiveness of the intrusion detection system, key features are identified and chosen in this step. Fuzzy logic is crucial in establishing the applicability and importance of each attribute. [21,22].

Feature Selection: The retrieved features occasionally could be high-dimensional or include extraneous data. The most informative and discriminative features for intrusion detection are found using feature selection approaches. As a result, the dimensionality is decreased and the detection process is made more efficient and effective [23,24].

Training and Testing: The data is split into training and testing datasets after feature extraction and selection. Using labeled samples of typical and intrusive network behavior, a machine learning model, such as a classification algorithm, is built using the training dataset. The trained model's performance is assessed using the testing dataset by looking at how well it can categorize unknown data [25].

Intrusion Detection: After the model has been trained and validated, it may be used to identify intrusions in fresh, unexplored data. The trained model predicts whether the observed behavior is typical or suggestive of an intrusion using the extracted features from incoming network data and system logs. Based on the outcomes of the detection, thresholds and decision

Peer Reviewed & Refereed Journal
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.46

rules can be established to send alerts or start additional activities [26].

IV. SUPPORT VECTOR CLASSIFICATION

The selected features are used as input to the support vector classification algorithm. By utilizing the knowledge from the fuzzy logic-based feature selection, the support vector classifier can more effectively distinguish between normal and intrusive network activities [27].

An approach for machine learning that can be used to detect intrusions in the cloud is called Support Vector Classification (SVC). Support vector machines (SVMs) are a type of supervised learning technique called SVC. It employs labeled training examples to categorize data into various group[28]s.

Computing the SVM for classification: we can denote the optimization problem and the solution of the optimization problem in a special way, that only involved the input feature via inner product. now this transform directly to the feature vector h(x)

Classification: Thelinearlogisticregressionmodelisused

$$: \Pr(Y = 1|x, \theta j\ell) = 11 + e - \theta Tj\ell x.$$

The LaGrange dual function has the form

$$LD = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N} \alpha_i\alpha_i' y_i y_i' \left(h(x_i)h(x_i')\right)$$

Now the solution function f(x) can be written

$$f(x) = h(x)^T\beta + \beta_0$$

$$= \sum_{i=1}^{N} \alpha_i y_i \left(h(x), h(x_i)\right) + \beta_0$$

As before α_i, β_0 can be determine by solving yi, f(xi) =1 for all Xi

We need not specify the transformation of h(x) , but require only knowledge of the kernel function[29,30]. K(x,x')=(h(x),h(x'))

That compute the inner product in the transform space . k should be symmetric positive

Three popular choice for K in the SVM

Dth degree polynomial $k(x, x') = \left(1 + (x, x')\right)^d$

Radial bias= $k(x, x') = \exp(-\gamma\|x - x'\|^2)$

Neural network $k(x, x') = \tanh(k_1(x, x') + k_2)$

Consider, for instance, a feature space with two inputs (X1 and X2) and a degree 2 apolynomila

$$K(x, x') = [(1 + (x_1x'])^2$$

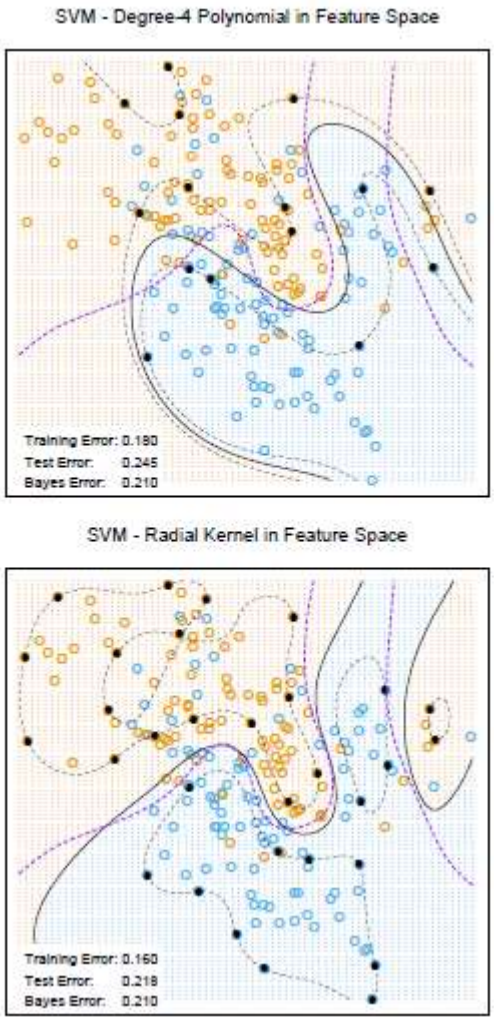
$$= (1 + x_1x_1' + x_2x_2')^2$$

$$= 1 + 2x_1x_1' + 2x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_1'x_2x_2'$$

the M=6 and if we choose h1(X)=1 h2(X)=root X1X2,

Then K(X,X')=(h(X),h'(X'))

$$\hat{f}(x) = \sum_{i=1}^{N} \alpha_i y_i k(x, x_i) + \hat{\beta}_0$$



Dataset Preparation: Gather and prepare cloud environment's raw network data and system logs. Extract pertinent information from the data, such as user behavior, system resource utilization, network traffic statistics, and other pertinent attributes. Create a labeled dataset where each instance is linked to a class label that denotes whether it represents normal or intrusive activity.

Feature Selection and Normalization: Selecting features for intrusion detection can be done by performing feature selection. The goal of this stage is to reduce dimensionality and concentrate on important qualities. Additionally, standardize the feature values to make sure that scaling for various features are comparable, which can enhance SVC performance.

Dataset Split: Make two subsets out of the labeled dataset: a training set and a testing set. The SVC model is trained using the training set, and its performance on untried data is assessed using the testing set.

Model Training: On the training set, use the SVC method. With a maximal margin of separation between the data points of various classes, SVC builds a hyperplane. Support vectors, the nearest data points to the

Peer Reviewed & Refereed Journal
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.46

decision boundary, are used to calculate the hyperplane[31].

Model Evaluation: Utilize the testing set to assess the SVC model's performance after training. Establish metrics for evaluating the model's capacity to distinguish between normal and intrusive behavior, such as accuracy, precision, recall, and F1 score. For the purpose of improving the performance of the model, cross-validation techniques can be used to modify the hyperparameters of SVC, such as the regularization parameter and kernel type[32].

Intrusion Detection: The SVC model can be used to process fresh, unexplored data from the cloud after being trained and validated. The trained SVC model predicts whether the behavior is typical or suggestive of an intrusion using the features retrieved from network data and system logs. It is possible to take the proper steps based on the predictions of the model, such as creating alerts, preventing suspicious activity, or starting a more thorough inquiry [33].

V. RELATED WORK

The "Advanced IDS Management Architecture" authors [2] proposed an IDS that combines an event gatherer and the Virtual Machine Monitor (VMM). An integrated control panel and various sensors make up this IDS. The Event Gatherer plugin performs the roles of handler, sender, and receiver to enable the integration of several sensors. This architecture makes use of IDMEF (Intrusion Detection Message Exchange Format). Through an interface, users can view the results reports. [34]

When a user wants to access the cloud, AAA examines the user's credentials and determines the most recent anomalous level based on that. Three categories of security exist: high, medium, and low.

A multi-threaded NIDS designed to function in a distributed cloud environment has been proposed by Gul and Hussain [4]. Three components make up this multi-threaded NIDS: the capture and queuing, analysis/processing, and reporting modules. The network packets must be read by the capture module before being sent to the shared queue for analysis.

The authors presented a methodology for integrating a network intrusion detection system (NIDS) in the Cloud in [5]. Snort and the signature apriorism algorithm, which can generate new rules from intercepted packets, are included in the proposed NIDS module. [35].

Table 1: Synthesis of the related work

IDS/Features	IDS management architecture that is advanced	a system for detecting cloud intrusions	technique for preventing and detecting clouds	a better hybrid IDS
Type	Collaborative	Collaborative	Intelligent	Intelligent
the capacity to recognize unidentified attacks	No	No	Could be	Could be
the capability of content analysis of encrypted streams	Yes	Yes	No	Yes
Encrypting warnings that are exchanged	No	No	No	No
spread of identified attacks	No	Alert system message usage	No	No

Intelligent Intrusion Detection Model for Virtual Machine

The foundation of cloud computing is virtualization, which refers to the deployment of various services and applications inside of virtual computers. It is crucial to implement intrusion detection systems in the Cloud at the virtual machine level because virtual machines use a variety of operating systems (OS) and expose a range of applications to cloud customers. As a result, any vulnerability in these systems and applications can be remotely exploited by hackers.

We suggest a smart host-based intrusion detection system (SHIDS) as a security tool to monitor the hypervisor and virtual machines on that hypervisor, detect malicious activities at the VM level, and guard against attacks targeting apps running on virtual machines. This approach offers many advantages in terms of portability and costs. The proposed SHIDS behaves like a HIDS and controls the state of the virtual machine since it has access to its stored information, whether it be in RAM, the file system, log files, or audit trails. The virtual machine is outfitted with a SHIDS (Smart Host-Based Intrusion Detection System), which is a customized HIDS (Host Based Intrusion Detection

Peer Reviewed & Refereed Journal
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.46

System) designed to support data mining in order to detect unknown attacks. The SHIDS can also examine every action that takes place on the virtual machine that is hosting the cloud services [8].

Since SHIDS has access to the encryption keys and certificates on the computer where it is installed, it can examine the content of encrypted streams but NIDS (Network-based intrusion detection system) cannot. While "Trojan" attacks are challenging for NIDS to identify, SHIDS can do so with ease.

VI. VIRTUAL MACHINE COMPONENTS

Is the operating system software, such as Linux or Windows, that controls the resources of computers' hardware and applications.

Utilizer software It is an application that the cloud offers and is made specifically for its users.

An intrusion detection system that is installed on a single host machine is known as a SHIDS (Smart Host Based Intrusion Detection System). It is a HIDS that has been upgraded and enhanced to support and integrate data mining and machine learning modules in order to detect new and undiscovered assaults that are not recorded in the IDS database. [12].

VII. SHIDS ARCHITECTURE

The SHIDS's many parts are as follows:

The module responsible for recording alerts, storing them in the database, and exchanging them with other intrusion detection systems is the logging and alert system.

IDS database: The signatures of the numerous attacks that have been previously identified are kept in a MySQL database.

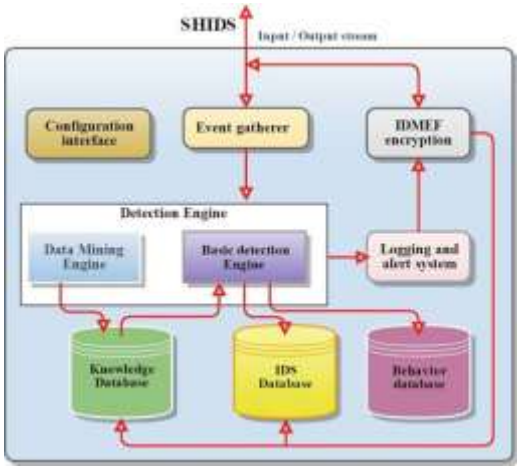
Behavior database: This is a database used to keep track of prior user behavior[15].

Interface for configuration: This is the interface used to set up and parameterize the SHIDS.

Data Mining Engine: This tool enables the use of data mining techniques like statistics, frequent pattern mining, clustering, and classification to identify previously unidentified attacks.

Knowledge base: This is a database that houses threats that have been discovered by the data mining engine after using various data mining techniques.

On a host system, the SHIDS acts like a demon or a regular service to look for suspicious activity. The SHIDS can keep an eye on user activity, system activity, and harmful activity like Trojans, viruses, and worms. The basic detection engine is a customized program responsible for analyzing the events and identifying attacks based on the signatures contained in the IDS database. The event gatherer, which gathers all events coming from outside to the virtual machine, sends the coming streams to the basic detection engine. The basic detection engine searches the database for a match when an input is received, and we added a behavior database using an anomaly detection technique based on statistical measures to enhance the SHIDS. This technique focuses on characterizing past behavior of specific users or groups of users to find significant deviations. Numerous parameters, including the user's connection time, the amount of password failures, and CPU and memory use, are taken in account. The basic detection engine will then contact the data mining engine, which will use data mining techniques like clustering, also-citation, and fuzzy logic to see if the received signature matches an intrusion. If an intrusion is found, it will be recorded in the knowledge base, and the basic detection engine will be notified. If no matching is found, this means that the signature is not stored in the database. In order to update the other SHIDS, the logging and alarm system sends the detected attack to the central ids in an IDMEF format, as illustrated in Fig. 3.



Handled. RFC 2007 describes the format's specifics in depth. using RFC 4765 presents a DTD and implementation of the XML data model. In Fig. 4, an illustration of an IDMEF message is displayed.

The extensibility and openness of the IDMEF standard led to its selection. IDMEF message types include alerts and heartbeats. Based on the IDMEF XML schema supplied by the RFC, the IDMEF library is created by JAXB. A class and its members are generated by JAXB based on a specific XML Java mapping. Ring-tools (Hardware Random Generator) was used to prevent users from sniffing traffic from insiders while RSA encryption was used to secure the exchanges between the central agent and the other SHIDSs in order to prevent anyone from reading the IDMEF alerts on the network[22].

VIII. THE SUGGESTED STRUCTURE

To secure the cloud environment, there are two possible scenarios. The first is to set up a distributed intrusion detection architecture in which each HIDS

Peer Reviewed & Refereed Journal
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.46

communicates by exchanging IDMEF alerts. In this case, if an intrusion is discovered by one SHIDS, it will be communicated to all other SHIDS so they can update their databases, as shown in Fig. 5. The disadvantage of this architecture is that it will increase network traffic because SHIDS exchange alerts frequently. Additionally, if the number of servers is high, this will result in a large number of alerts being generated, which will reduce SHIDS performance because the alerts come from various sources[25].

The second scenario is based on a centralized IDS architecture (Fig. 6) that is based on the principle of collaboration between many SHIDS deployed on the different virtual machines (assuming that we have n virtual machines VM1, VM2 VM and m. physical apparatus This technique has numerous advantages in terms of portability and costs. PM1, PM2,... PM with m # n) in the cloud to detect and protect against threats targeting apps operating on these virtual machines. The idea behind this strategy is that the several SHIDSs are installed in each VM and work together by exchanging alerts regarding detected intrusions. A central agent in our model is in charge of receiving notifications from the other SHIDS as well as writing to and reading from a central database to synchronize the local databases of each SHIDS. This idea synchronizes the several SHIDS, and the central agent system notifies nearby neighbors of any attacks it detects.

We chose the second scenario's architectural design since it provides more advantages than the model suggested in scenario one.

Mathematical Equation for Instruction detection in cloud Instruction detection in the cloud typically involves the application of various techniques and algorithms rather than a single mathematical equation[27].

Let's consider a basic mathematical framework for instruction detection:

Features Extraction

Let X be the input data matrix, where each row represents a data sample and each column represents a feature.

$X = [x_1, x_2, x_3, ..., xn]$, where n is the number of features.

ModelTraining:

Let Y be the target variable that indicates whether an instruction is normal or intrusive.

$Y = [y_1, y_2, y_3, ..., yn]$, where each yi corresponds to the label for the corresponding data sample in X.

Build a classification model using a specific algorithm (e.g., Support Vector Machines, Random Forest, Neural Networks).

The model can be represented as $f(X) = \hat{Y}$, where $\hat{Y}$ represents the predicted labels.

Prediction and Decision:

Given a new data sample x, the trained model can predict its label ŷ.

$\hat{y} = f(x)$.

Thresholding and Decision Rules

Based on the predicted label ŷ and a predefined threshold, a decision rule can be applied to determine if the instruction is classified as normal or intrusive. For example, if $\hat{y} \geq$ threshold, the instruction is classified as intrusive; otherwise, it is classified as normal. It's important to note that the mathematical equations and formulas may vary depending on the specific algorithm and techniques used for instruction detection.

Different algorithms have their own mathematical models, such as the equations defining support vector machines or decision trees. Additionally, advanced techniques like deep learning models involve complex mathematical operations, such as matrix multiplications, activation functions, and loss functions. However, providing a comprehensive mathematical equation for instruction detection in the cloud would require specifying a particular algorithm or model.

IX. RESULT

Using network simulator 2, the network's performance is evaluated in three different scenarios, including attacks from black holes and gray holes. Throughput based on entropy fluctuation, packet delivery ratio based on entropy variation, and jitter are three performance indicators used to assess performance. Node Source Destination Node Intermediate Node IPS Methodology Check the AODV Node type

Based on the findings, we arrive at the following conclusions, which are evident in table 3. Under IPS, the packet delivery ratio stays constant whether there are assaults or not. Despite attacks being present, the intrusion mechanism system ensures that no packets are dropped.

PARAMETERS	NOT WITH ATTACKS	WITH ASSAULTS	INCLUDING ATTACK UNDER IPS
Ratio of packets delivered	1.000	0.606	1.000
Packet loss	0	24	0
Jitter	7.86	6.79	11.21

X. CONCLUSION

The suggested approach not only recognizes the attack but also its range and extension. By utilizing fuzzy logic, the proposed method offers a noble answer and makes it easier to spot an assault. Additionally, the system includes an IPS mechanism technique that uses input from fuzzy technology and offers safe data transmission across the network. The traffic of black hole and gray hole attacks is likewise monitored by IPS. When compared to existing methods, the outcome clearly shown that this method identifies the attack in an effective manner. Future work will focus on reducing the

Peer Reviewed & Refereed Journal
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.46

jitter value, which increases when an IPS is present and is caused by route alteration when there are attacks.

REFERENCES

[1] Enache, A. C. & Patriciu, V. V. (2014, May). Intrusions detection based on support vector machine optimized with swarm intelligence. In: *IEEE 9th IEEE international symposium on applied computational intelligence and informatics (SACI)*, pp. 153-158. IEEE.

[2] Husain, M. S. (2020). A review of information security from consumer's perspective especially in online transactions. *International Journal of Engineering and Management Research, 10*.

[3] Teng, S., Du, H., Wu, N., Zhang, W. & Su, J. (2010). A cooperative network intrusion detection based on fuzzy SVMs. *J. Networks, 5*(4), 475-483.

[4] Shakeel, N., Haroon, M. & Ahmad, F. (2021). A study of wsn and analysis of packet drop during transmission. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*.

[5] Siddiqui, Z. A. & Haroon, M. (2023). Research on significant factors affecting adoption of blockchain technology for enterprise distributed applications based on integrated MCDM FCEM- MULTIMOORA- FG method. *Engineering Applications of Artificial Intelligence, 118*, 105699.

[6] Teng, S., Du, H., Wu, N., Zhang, W. & Su, J. (2010). A cooperative network intrusion detection based on fuzzy SVMs. *J. Networks, 5*(4), 475-483.

[7] Khan, N. & Haroon, M. (2023). A personalized tour recommender in python using decision tree. *International Journal of Engineering and Management Research, 13*(3), 168-174.

[8] Salama, M. A., Eid, H. F., Ramadan, R. A., Darwish, A. & Hassanien, A. E. (2011). Hybrid intelligent intrusion detection scheme. In: *Soft computing in industrial applications*, pp. 293-303. Springer Berlin Heidelberg.

[9] Khan, W. & Haroon, M. (2022). A pilot study and survey on methods for anomaly detection in online social networks. In: *Human-Centric Smart Computing: Proceedings of ICHCSC*, pp. 119-128. Singapore: Springer Nature.

[10] Khan, W. (2021). An exhaustive review on state-of-the-art techniques for anomaly detection on attributed networks. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12*(10), 6707-6722.

[11] Husain, M. S. & Haroon, D. M. (2020). An enriched information security framework from various attacks in the IoT. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*.

[12] Khan, W. & Haroon, M. (2022). An efficient framework for anomaly detection in attributed social networks. *International Journal of Information Technology, 14*(6), 3069-3076.

[13] Khan, W. & Haroon, M. (2022). An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks. *International Journal of Cognitive Computing in Engineering, 3*, 153-160.

[14] Zhao, R., Yin, Y., Shi, Y. & Xue, Z. (2020). Intelligent intrusion detection based on federated learning aided long short-term memory. *Physical Communication, 42*, 101157.

[15] Srivastava, S., Haroon, M. & Bajaj, A. (2013, September). Web document information extraction using class attribute approach. In: *4th International Conference on Computer and Communication Technology (ICCCT)*, pp. 17-22. IEEE.

[16] Tripathi, M. M., Haroon, M., Khan, Z. & Husain, M. S. (2022). *Security in digital healthcare system. pervasive healthcare: A compendium of critical factors for success*, 217-231.

[17] Khan, W., Haroon, M., Khan, A. N., Hasan, M. K., Khan, A., Mokhtar, U. A. & Islam, S. (2022). DVAEGMM: Dual variational autoencoder with gaussian mixture model for anomaly detection on attributed networks. *IEEE Access, 10*, 91160-91176.

[18] Haroon, M., Tripathi, M. M. & Ahmad, F. (2020). Application of machine learning in forensic science. In: *Critical Concepts, Standards, and Techniques in Cyber Forensics*, pp. 228-239. IGI Global.

[19] Husain, M. S. (2020). A review of information security from consumer's perspective especially in online transactions. *International Journal of Engineering and Management Research, 10*.

[20] Zhang, H., Li, Y., Lv, Z., Sangaiah, A. K. & Huang, T. (2020). A real-time and ubiquitous network attack detection based on deep belief network and support vector machine. *IEEE/CAA Journal of Automatica Sinica, 7*(3), 790-799.

[21] Li, L. & Zhao, K. N. (2011, May). A new intrusion detection system based on rough set theory and fuzzy support vector machine. In: *3rd International Workshop on Intelligent Systems and Applications*, pp. 1-5. IEEE.

[22] Tripathi, M. M., Haroon, M. & Ahmad, F. (2022). A survey on multimedia technology and internet of things. *Multimedia Technologies in the Internet of Things Environment, 2*, 69-87.

[23] Scholar, P. G. (2021). *Satiating a user-delineated time constraints while scheduling workflow in cloud environments*.

[24] Ma, T., Yu, Y., Wang, F., Zhang, Q. & Chen, X. (2018). A hybrid methodologies for intrusion detection based deep neural network with support vector machine and clustering technique. In: *Frontier Computing: Theory, Technologies and Applications FC 20165*, pp. 123-134. Springer Singapore.

[25] Haroon, M., Misra, D. K., Husain, M., Tripathi, M. M. & Khan, A. (2023). Security issues in the internet of things for the development of smart cities. In: *Advances in Cyberology and the Advent of the Next-Gen Information Revolution*, pp. 123-137. IGI Global.

[26] Ren, W., Cao, J. & Wu, X. (2009, November). Application of network intrusion detection based on fuzzy c-means clustering algorithm. In: *Third International Symposium on Intelligent Information Technology Application*, *3*, pp. 19-22. IEEE.

[27] Enache, A. C. & Sgarciu, V. (2014, October). Anomaly intrusions detection based on support vector machines with bat algorithm. In: *18th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 856-861. IEEE.

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

[29] Koshal, J. & Bag, M. (2012). Cascading of C4. 5 decision tree and support vector machine for rule based intrusion detection system. *International Journal of Computer Network and Information Security*, *4*(8), 8.

[30] Haroon, M., Tripathi, M. M., Ahmad, T. & Afsaruddin. (2022). Improving the healthcare and public health critical infrastructure by soft computing: An overview. *Pervasive Healthcare: A Compendium of Critical Factors for Success*, 59-71.

[31] Masarat, S., Taheri, H. & Sharifian, S. (2014, October). A novel framework, based on fuzzy ensemble of classifiers for intrusion detection systems. In: *4th international conference on computer and knowledge engineering (ICCKE)*, pp. 165-170. IEEE.

[32] Xie, Y. & Zhang, Y. (2012, August). An intelligent anomaly analysis for intrusion detection based on SVM. In: *International conference on computer science and information processing (CSIP)*, pp. 739-742. IEEE.

[33] Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P. & Kannan, A. (2013). Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. *EURASIP Journal on Wireless Communications and Networking*, *2013*, 1-16.