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

Gold Rush Optimization-Driven Random Forest Approach for Intrusion Detection System in Edge and Fog Computing Settings

[1]Dr. Sweta Kumari, [2]Shweta Singh, [3]Beemkumar Nagappan, [4]Satish Kumar Jangid, [5]Sachin Mittal

Submitted: 05/02/2024 Revised : 13/03/2024 Accepted: 21/03/2024

Abstract: Although both edge and fog computing architectures improve latency minimization and real-time data processing, their scattered nature presents significant security problems, particularly in intrusion detection. In this study, we provided an innovative gold rush optimization-driven random forest (GRO-RF) technique for effective intrusion detection systems (IDS). To examine the performance, the UNSW-NB15 public dataset is utilized to train the suggested GRO-RF technique. The z-score normalization approach is used to preprocess the raw samples to rearrange the data without noise and duplicates. To extract important features, the cleaned data is subjected to additional processing throughout the feature extraction process using linear discriminate analysis (LDA). The RF technique is used to identify intrusions in edge and fog computing environments using the retrieved data. GRO is designed to enhance the misclassification of RF by increasing accuracy and reducing the error rate in categorization. The suggested method is implemented using a Python program and its efficacy in detecting intrusions is evaluated against other current approaches using various metrics such as precision (98.45%), accuracy (99%), F-measure (96.89%) and recall (97.25%). We show that, in edge and fog computing scenarios, the GRO-RF technique has the highest intrusion detection accuracy compared to the other methods, based on the results of the experiment.

Keywords: *Edge computing, fog computing, Gold Rush Optimization-Driven Random Forest (GRO-RF), Intrusion Detection Systems (IDS), Security.*

1. Introduction

Intrusion detection systems (IDS) are essential defensive mechanisms in the ecology of computer security. The foundation of IDS is the theory that regular user behavior differs from intruders.

1.1. Edge Computing

The Internet of Things (IoT) and cloud computing are being extended to the edge of the network by the emerging edge computing trend. An IDS is employed in edge computing, comparable to most other systems, to reduce cyber security concerns. It is difficult to allocate resources inside IDS effectively and equitably since edge nodes have constraints. In general, accessibility, geo-distribution, positional awareness and latency are supported by IoT applications. It has been determined that edge computing is a workable solution to cloud computing shortcomings in enabling IoT applications.

Edge computing makes it easier to provide users of networks with enhanced safety network communication, more flexible access and reduced latency services. However, there are a lot of difficulties with network security, performance and stability. Edge computing networks are subject to several threats. The harm that malicious assaults could cause if appropriate security and privacy protection mechanisms are not in place offsets the advantages of edge computing. An IDS is a vital safety tool that can identify network intrusions and security threats. A significant portion of user service requests in edge computing are handled by edge nodes. However, edge nodes lack the resources necessary to carry out thorough detections. A significant challenge is to allocate resources among edge nodes efficiently.

1.2. Fog computing

Fog computing offer processing, storage as well as application services to end users, along with the connectivity between fog nodes and cloud servers. Fog computing has several security and safety problems as a result of its adoption in various areas with lax security. The detection of misuse and anomaly detection are the two main types of IDS. Misuse detection, often referred

1Assistant Professor, Department of ISME, ATLAS SkillTech University, Mumbai, Maharashtra, India, Email Id- sweta.kumari@atlasuniversity.edu.in, Orcid Id- 0000-0003-3173-1140
2Assistant Professor, Maharishi School of Engineering & Technology, Maharishi University of Information Technology, Lucknow, India -226036, Email Id- shwetasingh580@gmail.com, Orcid Id- 0000-0001-54589269
3Professor, Department of Mechanical Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Karnataka - 562112, India, Email Id- n.beemkumar@jainuniversity.ac.in, Orcid Id- 0000-0003-3868-0382
4Assistant Professor, Department of Electrical Engineering, Vivekananda Global University, Jaipur jangid.satish@vgu.ac.in, Orcid Id- 0000-0003-3506-8833
5Centre of Research Impact and Outcome,Chitkara University, Rajpura-140417, Punjab, India, Email ID: sachin.mittal.orp@chitkara.edu.in, Orcid Id- https://orcid.org/0009-0006-7510-6725

as signature-based detection, mostly utilizes the guidelines that the network administrator has established. Therefore it can only identify known attacks and ignore newly developed ones. Anomaly detection, the second category, employs a statistical method to identify novel and unidentified assaults, which allows it to uncover issues with the initial one. Nevertheless, the majority of established anomaly detectors have several difficulties, including a high incidence of false positive alerts and costly calculations. As a result, the majority of IDS available exclusively employ abuse detection. Therefore, to create IDS, users need to understand the way attacks happen, gather data, set up remote and local access, as well as initiate an attack. The purpose of the study is to detect the efficacy in IDS for edge and fog computing using the GRO-RF technique.

The following parts: An overview of related works is given in part 2, a more thorough explanation of the methods is given in part 3 as well as discussion and results are presented in part 4. Part 5 provides conclusion and offers suggestions for more research.

2. Related works

The study [6] suggested using Long Short-Term Memory networks (LSTM) and convolutional neural networks (CNN) in an intrusion classification framework utilizing the benefits of deep learning (DL) techniques for precise attack prediction. The study [7] represented that individual immune, lightweight IDS to the layer of fog were based on anomalies. The suggested architecture divided the IDS functionalities among the cloud and fog nodes to achieve minimal resource overhead. The study [8] developed a decentralized version of the current cloud-based security architecture that was based on anomaly detection and it was built on local fog nodes. The study [9] provided novel IDS that were lightweight and distributed. The suggested IDS combine a variation Automatic encoder using many layers of perception to offer capable and exact IDS.

The study [10] presented an enhanced IDS model to classify IoT and Edge of Things (EoT) threats. Ten distinct machine learning models were used to provide an enhanced IDSs-IoT that protects IoT and EoT appliances and devices. The study [11] recommended developing an adaptive and robust network IDS to identify and categorize network threats using DL architectures. The focus was on how DL or deep neural networks (DNNs) could allow adaptive IDSs that can detect and identify novel or zero-day network behavioral characteristics. This can lead to the removal of system intruders and a decreased chance of compromise. The study [12] investigated the possibility of using ML classification methods to protect the IoT against denial-of-service (DoS) attacks. An extensive investigation was conducted on the classifiers that have the potential to progress the creation of IDS based on anomalies. The study [13] provided the characteristics of the adversarial issue in network IDS. They concentrate on the hit viewpoint that encompasses methods to produce hostile cases that can sidestep different ML models.

3. Methodology

The UNSW-NB15 dataset was initially acquired as part of the procedure. The collected datasets are preprocessed using z-score normalization. Linear discriminate analysis (LDA) is used as a feature extraction. The suggested GRO-RF is used for edge and fog computing using IDS. Fig 1 depicts the flow of the proposed methodology.

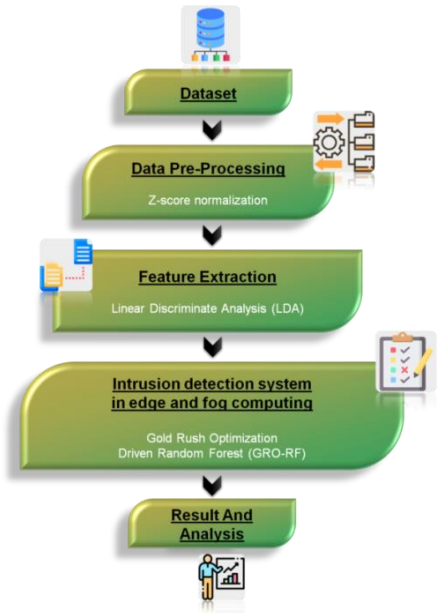

Fig 1. The sequence of the suggested methodology

3.1. Dataset

The dataset of UNSW-NB15 [14] strikes is used in our experimental procedures. “The following types of network attacks are present in instances of the UNSW-NB15: Backdoor, Exploits, Worms, Reconnaissance, Fuzzers, Generic, DoS, Analysis and Shell-code”.

The two primary datasets which together make the basis of UNSW-NB15 are UNSW-NB15-TEST (100%) and UNSW-NB15-TRAIN, which are used for preparing different models and testing the models that have been trained, respectively. The UNSW-NB15-TRAIN was further divided into two divisions in our work: the UNSW-NB15-TRAIN-1 (75%) was used for training and the UNSW-NB15-VAL (25%) was used for validation. The purpose of this second division is to serve as an assurance of sanity for the training process outcomes. Data leaking is a phenomenon that should be avoided while using this technique since training a model on an assessment or test set could result in leakage.

3.2. Data preprocessing using z-score normalization

Normalization is the term used to describe the Z-score parameter. The z-score parameter is used to standardize unstructured data using the following formulas:

$$P_g{}' = \frac{x_g - C}{std\ (C)} \quad (1)$$

Values of the x_g are Z-scores normalized to one. The value of the row in the gth column is x_g.

$$std(V) = \sqrt{\frac{1}{(a-1)}} \sum_{g=1}^{a} (d_v - \bar{V}\ )^2 \quad (2)$$

$$\bar{V} = \frac{1}{m} \sum_{v=1}^{a} d_v \quad (3)$$

The z-score normalization was utilized to recognize the normalized ones in each row. For example, all values for a row are set to zero if the standard deviation of that row is 0 and all weights are comparable. The z-score normalization is used to eliminate dataset abnormalities that result from transitive dependencies.

3.3. Feature extraction uisng Linear discriminate analysis (LDA)

After preprocessing technique the feature extraction method is used to detect features. An LDA lowers the issues' complexity, improves the standard deviation classifier's generalization and requires less processing time. A map is transformed from a d-dimensional field of view to an m-dimensional feature space using the transformation matrix.

$$V{:}S^u \rightarrow S^m m < u \quad (4)$$

By converting the resulting area into the involvement area linearly, the LDA could eliminate features. With a few little deviations in the end, the two classifiers are built identically. The following formula defines the length of the d within-class correlation square matrix.

$$U_v = \frac{1}{n} \sum_{j=1}^{2} \sum_{i=1}^{j} \left[(Y_i)_j - n_j\right] \left[(Y_i)_j - n_j\right]^d \quad (5)$$

When n_j, the group i signaling vector, is accessible. Information is separated into two categories when labeling an event: non-occurrence zones and occurrence regions.

$$C_O = \frac{n_1}{n}(m_1 - m)(m_1 - m)^c + \frac{n_2}{n}(m_2 - m)(m_2 - m)^c \quad (6)$$

The vector m represents the whole collection of variables. Developing a matrix of transformations that maximizes the variation in categories and balances the range across classes is the objective that looks regarding LDA.

$$W_P = \frac{m_1 m_2}{m^2}(n_2 - n_1)(n_2 - n_1)^v \quad (7)$$

The stage of $U_V^{-1} C_A$ is equivalent to one person, even if Y_C has a level that is inhabited and its opposite exists. To put it another way, there is only a single nonzero eigenvalue.

$$P_1 = \frac{U_V^{-1}(n_2 - n_1)}{\|U_V^{-1}(m_2 - m_1)\|} \quad (8)$$

The resultant vector V is produced by the transformation function and the corresponding eigenvector needs one attribute and can operate as a vector $m_2 - m_1$.

$$G = F_1^c G = (m_2 - m_1)^c U_V^{-1} \quad (9)$$

The effect of the generalized function is seen in Equation (9).

$$X = BY = V(n_2 - n_1)^s Y \quad (10)$$

Although LDA categorization is required in advance of inversion within-class invariance matrix Y_c for this to happen, it yields an unconditioned matrix.

$$U_{vJ} = U_v + \delta I \quad (11)$$

Distance-based algorithms benefit from normalization, which helps to deal with outliers since it makes the measurements more consistent.

3.4. Intrusion Detection System in Edge and Fog Computing using Gold Rush Optimization-Driven Random Forest (GRO-RF)

The random forest (RF) classification uses many decision trees to build a forest. Large datasets optimize the performance of the RF method for edge and fog computing. Using a larger number of trees in the decision-making process improves the forest. Every stage of a tree's formation considers a leaf's growth. Two identically structured halves of the dataset are randomly selected. The algorithm's margin function can be represented as:

$$nh(W, Z) - \alpha v_l J(W) = Z) - \max_{i \neq Z} \alpha v_l J(g_l(W) = i) \quad (12)$$

If $X\ and\ W$ are random vectors, $J(\cdot)$ is the indicator function and $g_1(W), g_2(W), \ldots, and\ g_2(w)$ are the ensemble of classifiers. The mistake is provided by:

$$OF^* = O_{W,Z}(nh(W, Z) < 0) \quad (13)$$

$$QE, g_l(W) = g(W,\ \Theta_l) \quad (14)$$

The following formula provides the margin function for an RF.

$$nq(W, Z) = O_\Theta(g(W, \Theta) = Z) - \max_{i \neq Z} O_\Theta(g(W, \Theta) = i) \quad (15)$$

Where the classifier $\{g(W, \Theta)\}$'s strength is determined by:

$$t = F_{W,Z}nq(W,Z) \quad (16)$$

To create a single decision tree with the maximum prediction accuracy, the RF algorithm combines many decision trees into one effective model for edge and fog computing. Fig 2 represents the functionality of the RF classifiers.

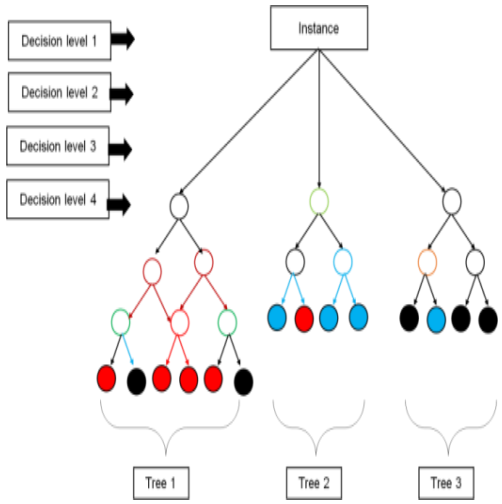

Fig 2. Functionality of RF classifier

The following stages outline the RF algorithm:

Step 1: Starting with a given data set C_1 with a n×m matrix, an additional dataset C_2 is generated by sampling and deleting IDS's one-third of the row data from the original data.

Step 2: After that is trained, the model creates a new dataset using the smaller samples and calculates the unbiased error in IDS.

Step 3: From the total number of columns (m), m_1 is chosen at every node point throughout the data collection in IDS.

Step 4: Several trees develop at the same time in this technique and the gathering of individual judgments in IDS for edge and fog computing is used to make the final prediction to maximize classification accuracy.

The Gold Rush Optimization (GRO) is optimized with RF to provide enhanced detection for IDS in edge and fog computing. The foundation of a Gold Rush Optimization (GRO) algorithm is the ability of human reasoning and judgment. The population-based evolutionary algorithm known as the GRO algorithm converges more quickly compared to other optimization methods. The location of the gold is to be found. Initially, a group of people are referred as operators station themselves at a random spot inside the search region. Each operator searches for gold using a metal detector. The operators walk as a group throughout each step, listening to the audio until they detect an increase in volume, at which point they halt. Each operator needs to keep an ear out for any noises made by other equipment and continuously check to determine if any of them are louder than the others. The group advances to the location with the greatest volume at each step. The exact position of the gold is found at the end. The possibility of going in the direction of the greatest volume or away from it is indicated by the three factors, α, β and γ. In the interval [0–1], the values for α, β and γ are chosen.

Level 1: Activation

In the search space, each operator occupies a random location as shown by Equation (17). A domain's upper and lower boundaries are represented by the numbers ka_j and wa_j (search space). N is the total amount of operators and rand is a random integer in the range [0– 1].

$$location_j^{(0)} = ka_j + (wa_j - ka_j) * rand, j = 1,2..,M \quad (17)$$

Level 2: Observing and selecting the ideal areas

SOP is a successful operator that locates the best spot. It is necessary to produce an SOP in this phase. Every iteration should conclude with the selection and retention of the top 10% of operators in the SOP.

Level 3: Distance-fitness

Equation (18) is used to compute the operator with the greatest chance of extracting gold, which is the analysis of each sound's loudness:

$$rate\ (j) = \frac{C_j}{\rho} * \frac{soud(higestvolume) - sound(j)}{soud(higestvolume) - sound(lowest\ volume) + \varepsilon} \quad (18)$$

To prevent singularities, the epsilon (ε) is a tiny positive integer. The parameters, ρ and C_j, shown in Equation (19) are employed to guard against environmental mistakes. The two operators' present positions are shown by the indices j and i.

$$\rho = 2 - \frac{iter}{max_{iter}}, C_j = \sqrt{(w_j - w_i)^2 + (z_j - z_i) + \cdots} \quad (19)$$

Level 4: Think – Decide - Act

Each operator produces distinct choices in this stage, each based on a combination of sounds shown in Equation (20).

$$new\ location(j) = location(j) + nc \times [(rate\ (i))rand] \quad (20)$$

The move direction coefficients nc are derived from Equation (21):

$$nc = \begin{cases} +1 \Rightarrow towards\ the\ loudest\ sound?\ \alpha > rand \\ -1 \Rightarrow away\ from\ the\ loudest\ sound?\ \alpha < rand \end{cases} \quad (21)$$

Level 5: Accurate location

If the position determined by Equation (20) is not inside the bounds of the issue, new locations are generated by using Equation (22). Coefficients β and γ are chosen so that $0< \beta < \gamma <1$.

$$new\ location\ (j) = \begin{cases} choose\ a\ neighboring\ location\ rand < \beta \\ select\ a\ new\ location\ randomly\ \beta < rand < \gamma \\ do\ not\ move\ \gamma < rand \end{cases} \quad (22)$$

Level 6: Ending

Eventually, the ending criteria are satisfied and steps 4 through 6 are repeated in a loop. Algorithm 1 depicts the process of GRO.

Algorithm 1: GRO

Establish the population of gold prospectors $W_j, j = 1,2, \dots, M$

Set up the new positions for the gold prospectors $W_{new_j} = W_j, j = 1,2, \dots, M$

Start s, k_1, k_2

W^* *is the most effective search engine*

Whereas $s \geq$ *the greatest amount of trials do*

For Every search engine representative j *do*

Determine the present search agent's suitability for the new role W_{new_j}

Change the current searching agent's location to W_j

Update your preferred search engine W^*

End

Modernize k_1, k_2

For every search engine representative j *do*

Determine the search agent's next position W_{new_j} *with an individual of the voyage, removal, or cooperation methods*

End

$s \leftarrow s + 1;$

Final return W^*

By improving reliability and decreasing the frequency of errors in categorization, the obtained data is combined with the GRO-RF approach to detect intrusions in edge and fog computing environments.

4. Result and discussion

The efficiency of the suggested and present techniques is evaluated using appropriate performance measures. Precision, F-measure, recall and accuracy are among the measures. Comparing the suggested GRO-RF technique's performance to that of different approaches, including support vector machine (SVM) [15], Logistic regression (LR) [16] and extreme gradient boosting (XG-Boost) [17], an analysis is conducted.

4.1. Accuracy

The model that forecasts the outcomes of real system operations or planned actions is compared to assess the accuracy. The proportion of values that are properly categorized is found using the accuracy in IDS. The accuracy of the suggested system is depicted in Table 1 and Fig 3. To compare the existing approach SVM attained 94%, LR attained 95% and XG-Boost attained 97%, both get below the desired 99% accuracy for GRO-RF.

Table 1. Quantitative outcomes of accuracy

Methods	**Accuracy (%)**
SVM [15]	94
LR [16]	95
XGBoost [17]	97
GRO-RF [Proposed]	99

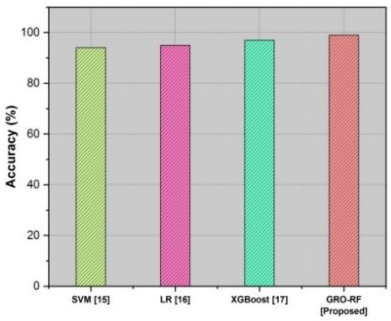

Fig 3. Comparison of accuracy

4.2. Precision

The accuracy rate of identifications is represented by the precision metric. The model's efficacy in classifying positive values is determined by IDS precision. The precision of the suggested system is depicted in Table 2 and Fig 4. To compare the existing approach SVM attained 95%, LR attained 97% and XG-Boost attained 94%, both get below the desired 98.45% precision for GRO-RF.

Methods	Precision (%)
SVM [15]	95
LR [16]	97
XGBoost [17]	94
GRO-RF [Proposed]	98.45

Table 2. Quantitative outcomes of precision

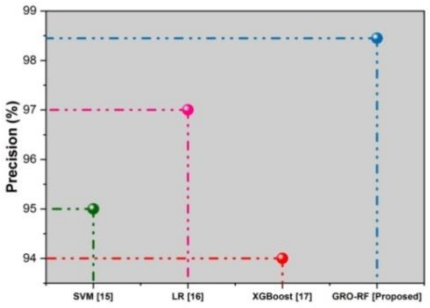

Fig 4. Comparison of precision

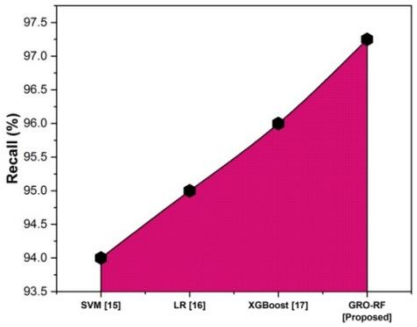

Fig 5. Comparison of recall

4.3. Recall

The recall is the percentage of relevant cases that could be located. It is employed to determine the extent to which the model can forecast positive values in IDS. The recall of the suggested system is depicted in Table 3 and Fig 5. When comparing the existing approach SVM attained 94%, LR attained 95% and XG-Boost attained 96%, both get below the desired 97.25% recall for GRO-RF.

Table 3. Quantitative outcomes of recall

Methods	Recall (%)
SVM [15]	94
LR [16]	95
XGBoost [17]	96
GRO-RF [Proposed]	97.25

4.4. F-measure

F-measure is defined as the harmonic mean of accuracy. By measuring the F-measure parameter in IDS, classification results can be made accurate. The f-measure of the suggested system is depicted in Table 4 and Fig 6. When compare the existing approach SVM attained 95%, LR attained 96% and XG-Boost attained 94%, both get below the desired 96.89% f-measure for GRO-RF.

Table 4. Quantitative outcomes of f-measure

Methods	F-measure (%)
SVM [15]	95
LR [16]	96
XGBoost [17]	94
GRO-RF [Proposed]	96.89

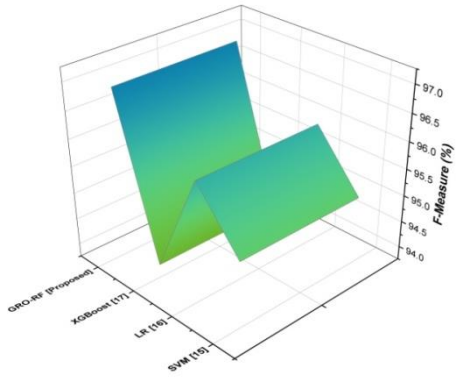

Fig 6. Comparison of f-measure

4.5. Discussion

Optimizing the IDS model's parameters is one of SVM's [15] main shortcomings. Usually, several parameters affect the SVM algorithm. One of LR's [16] drawbacks in classification is that it struggles to function properly in situations when the training sets are unbalanced. XG-Boost's [17] drawbacks include its time-consuming, greedy method and the fact that multi-threaded optimization is not required. In edge and fog computing, the GRO-RF method generates varying amounts of decision trees from multiple samples and uses their majority vote to determine the classification choice. GRO-RF has the advantage of allowing for greater accuracy without running the danger of IDS over-fitting.

5. Conclusion

The IDS are essential security elements in modern information technology-based enterprises. It could be difficult to provide an effective and high-performance IDS strategy for dealing with a broad range of security threats. The privacy and security features of cloud computing are two new concepts, edge computing and fog computing. A novel GRO-RF approach was developed in this publication to identify as well as categorize intrusions in fog and edge computing settings. The selected top attributes are used in the random forest approach to identify intrusions. GRO is designed to enhance inaccurate classification of RF by increasing accuracy and reducing the amount of error in categorization. In addition, a broad range of experimental studies with thorough simulations were conducted to demonstrate the improved results of the GRO-RF model. The experimental findings demonstrated the better performance of the suggested strategy in provisions of precision (98.45%), accuracy (99%), F-measure (96.89%) and recall (97.25%). IDS is unable to prevent or stop an attack that it has recently discovered by recognizing patterns in the attack or matching the attack's signature from the database. In further research, we plan to use an automated oversampling approach to enhance the minority class incidence in the UNSW-NB15 dataset during training.

References

[1] Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020). Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. IEEE Internet of Things Journal, 7(8), 6882-6897.

[2] Smys, S., Basar, A., & Wang, H. (2020). Hybrid intrusion detection system for internet of things (IoT). Journal of ISMAC, 2(04), 190-199.

[3] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

[4] Lopez-Martin, M., Carro, B., & Sanchez-Esguevillas, A. (2020). Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Systems with Applications, 141, 112963.

[5] Kasongo, S. M., & Sun, Y. (2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Computers & Security, 92, 101752.

[6] Kalaivani, K., & Chinnadurai, M. (2021). A Hybrid Deep Learning Intrusion Detection Model for Fog Computing Environment. Intelligent Automation & Soft Computing, 30(1).

[7] Aliyu, F., Sheltami, T., Deriche, M., & Nasser, N. (2022). Human immune-based intrusion detection and prevention system for fog computing. Journal of Network and Systems Management, 30, 1-27.

[8] Kumar, P., Gupta, G. P., & Tripathi, R. (2021). Design of anomaly-based intrusion detection system using fog computing for IoT network. Automatic Control and Computer Sciences, 55(2), 137-147.

[9] Labiod, Y., Amara Korba, A., & Ghoualmi, N. (2022). Fog computing-based intrusion detection architecture to protect iot networks. Wireless Personal Communications, 125(1), 231-259.

[10] Saheed, Y. K. (2022). Performance Improvement of Intrusion Detection System for Detecting Attacks on Internet of Things and Edge of Things. In Artificial Intelligence for Cloud and Edge Computing (pp. 321-339). Cham: Springer International Publishing.

[11] Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185, 239-247.

[12] Verma, A., & Ranga, V. (2020). Machine learning based intrusion detection systems for

IoT applications. Wireless Personal Communications, 111, 2287-2310.
[13] Alhajjar, E., Maxwell, P., & Bastian, N. (2021). Adversarial machine learning in network intrusion detection systems. Expert Systems with Applications, 186, 115782.
[14] Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data, 7, 1-20.
[15] Alqarni, A. A. (2023). Toward support-vector machine-based ant colony optimization algorithms for intrusion detection. Soft Computing, 27(10), 6297-6305.
[16] Bulso, N., Marsili, M., & Roudi, Y. (2019). On the complexity of logistic regression models. Neural computation, 31(8), 1592-1623.
[17] Zhang, P., Jia, Y., & Shang, Y. (2022). Research and application of XGBoost in imbalanced data. International Journal of Distributed Sensor Networks, 18(6), 15501329221106935.