Efficient Feature Generation with Modified Whale Optimization Algorithm to Classify the Intrusion Detection

Battini Sujatha[1], Sammulal Porika[2]

[1]Telangana Social Welfare Residential Degree College for women, Scholar of Computer science and Engg, JNTUH, Email:battinisujata@gmail.com
[2]Professor Department of computer science and Engg, JNTUH, College of Engineering, Email: sam@jntuh.ac.in

Received: 15.07.2024 Revised: 10.08.2024 Accepted: 06.09.2024

ABSTRACT
In recent years, cyberattacks and network intrusions have emerged as significant threats to applications that are connected to the Internet of Things (IoT). Existing methods for preventing and detecting intrusions are not capable of accurately identifying every sort of attack or irregularity in network data. This is because to a number of constraints. Researchers have also proposed a large number of strategies that are based on machine learning; nevertheless, the effectiveness of these strategies in terms of classification accuracy or multi-class categorization is restricted. Through the utilization of a number of algorithms for the purpose of processing and filtering data, this study presents a data-centric technique for the detection of irregularities and intrusions. Improving the quality of the training dataset is accomplished by the utilization of the FGen-MWO Algorithm, which stands for Feature Generation with Modified Whale Optimization. K-Means is an additional application of automated machine learning that is used to find the method with auto-tuned hyper-parameters that is suitable for the most accurate classification of data. Not only does this approach reduce the computational cost of run-time data assessment, but it also generates an optimal algorithm that does not require any human tweaks to the hyperparameters. Overperforming preceding algorithms by a significant margin, the algorithm that was developed as a result can handle a multi-class classification problem with an accuracy rate of 99.7%.

Keywords: Cyber-attacks: Network; Machine Learning; Intrusion; Accuracy and Anomalies.

INTRODUCTION

The Internet of Things (IoT) has become an integral element in various facets of contemporary life due to its integration of digital and physical components. Generally, components such as sensors and actuators are regarded as physical, whereas elements like transmission and storage medium are considered cyber. The Internet of Things necessitates a reduction in personnel for data reading and transfer, along with a diminished reliance on physical devices. IoT applications encompass Industry 4.0 [1], which mitigates equipment inefficiencies and enhances the overall quality of the network system. The Internet of Things (IoT) has been utilized for the control of smart household appliances, real-time data monitoring, and wearable gadgets that track vital indicators of individuals [2]. Additional IoT research domains encompass agriculture, healthcare, intelligent transportation, energy, and manufacturing.

Network security is increasingly vital for protecting the assets of private enterprises and individuals against intrusions, as more facets of their lives rely on the internet. Cyber-attacks are growing more intricate and challenging to avert. The yearly expenditure resulting from cyber-attacks, amounting to billions of dollars, is anticipated to increase in the forthcoming years. Monitoring and examination are crucial for identifying and averting network assaults in real-time. The primary line of defence for detecting breaches and safeguarding the network from invasions comprises antivirus software, access control, encryption, decryption, and firewalls, among other measures.

Nonetheless, these antiquated security techniques occasionally fail to safeguard the network against novel attack methodologies due to their inadequacy. Therefore, to effectively identify anomalous behaviour within the network, researchers are presently focusing on the development of robust intrusion detection systems (IDS) utilizing machine learning (ML) and deep learning (DL) methodologies. Artificial intelligence (AI) algorithms can diminish the necessity for human supervision while concurrently predicting and detecting various attacks, including novel ones, by recognizing patterns in generally obscured data. Data-driven methodologies can swiftly and efficiently identify problems, as the efficacy of both machine learning and deep learning models is contingent upon the dataset utilized for training. The

dataset is optimized, extraneous characteristics are removed in advance, and the data is processed to address missing values, as well as numerous and duplicated classes, due to the data-driven methodology. Data processing can eliminate the necessity for supplementary overhead associated with intricate feature extraction methods by removing all extraneous data features. This complements the assurance provided by [4].

The proposed model achieves data balance by removing instances of duplicate classes (under-sampling) and duplicating instances of redundant classes (over-sampling). Removing any attributes irrelevant to class identification enhances the dataset's quality and diminishes the computational resources needed. This is achieved by determining the contribution weight of each characteristic in class categorization through the Mutual Information (MI) index. Automatic machine learning (Auto-ML), such as K-Means, facilitates the assessment of an algorithm's performance in relation to minor modifications of its hyperparameters, hence augmenting the application of AI. Manually identifying and learning the optimal hyperparameters and methods is infeasible due to the necessity of boundless computations. AutoML has addressed this issue by converging to the global minimum of hyperparameters over multiple iterations. This study largely emphasizes enhancing data quality prior to classification to augment the classifier's performance. The primary contribution of the study is highlighted below.

- This study introduces the Feature Generation with Modified Whale Optimization (FGen-MWO) Algorithm to equilibrate the NSL-KDD dataset. This reduces bias in machine learning models, hence enhancing their overall performance.MI-based attribute collection involves selecting attributes that are more pertinent to the output class. MI evaluates the correlation of each feature in the dataset with the final class and eliminates features with low MI scores. Consequently, the dataset is reduced, thereby decreasing the training costs and durations of the machine learning models.
- The proposed study employs automated machine learning (AutoML) due to its efficacy in information processing for predicting the final classification. Furthermore, auto-ML optimizes the parameters of classification models by iterative execution until optimal results are achieved. Finally, the superior performance of the proposed framework is evidenced through comparison with other leading frameworks.
- A modified firefly-based optimization technique is employed to ascertain the mass density of user-centric data clusters and to train the categorized data for optimization alongside the salient features. The proposed combination strategy effectively decreases the detection error rate in the initial phase while enhancing data accuracy and sensitivity.

RELATED WORKS

Among the many fields that have found use for Internet of Things devices recently include healthcare, transportation, Industry 4.0, and others. Improving data quality is essential for information transmission, privacy assessments, and analysis; researchers have taken a data-driven strategy to achieve this goal. To reduce data transmission latency and extend sensor battery life, [5] advocated using a data-driven strategy in the Internet of Things.

A data-driven machine learning approach was suggested in [6] for the purpose of detecting lameness in cattle. A data-driven, Internet of Things (IoT)-connected, high-precision phase measurement device was introduced in [7]. To examine urban evolution patterns, a data-driven approach was developed, comprising the collecting information from various IoT sensors, which was then processed and interpreted by machine learning algorithms. An examination of the need for a data-driven approach to code analysis and cyber-security systems.

The proliferation of IoT devices and services has made intrusion detection systems (IDS) indispensable. An intrusion detection system (IDS) that is based on anomalies was conceived and constructed by [7] in order to improve the safety of Internet of Things edge devices. When creating intrusion detection systems (IDS) for IoT devices, researchers looked at how feature selection affected system efficiency and how important it was throughout development. Made an intrusion detection system for IoT systems that can classify things with about 89% accuracy. A hybrid intrusion detection system (IDS) with two levels that can detect threats and abnormalities in an internet of things (IoT) setting; it can also analyze and evaluate its own efficiency and detect any threats.

Intrusion detection systems that use binary and multi-class classification have been the subject of substantial research. A three-layer Multi-Layer Perceptron (MLP) model was suggested in the NSL-KDD dataset [10]. The proposed model achieved a multi-classification accuracy of 79.9% and a binary classification accuracy of 81.2%. Ibrahim et al. (2013) announced a new method using Self-Organizing Maps (SOM) that was 75.49% accurate at binary classification. Database cross-validation across decades. That's 95.7% success in binary classification for this study using the standard MLP. In addition, they assessed their proposed semi-supervised learning model by employing the NSL-KDD dataset. Fuzzy and

ensemble learning methods were employed in the model's construction. At 84.54%, the recommended approach was spot on. In [11], the Deep Belief Network (DBN) was described. It was based on the Restricted Boltzmann Machine (RBM) and had a softmax output layer for sorting things into multiple groups.

When tested on 10% of the KDD99 dataset, the proposed plan got a score of 98%. Tang et al. (2016) introduced the Deep Neural Network (DNN), which finds strange things by using Software Defined Networking (SDN). The suggested DNN was a three-layer network that had been trained using the NSL-KDD dataset. The binary classification achieved a 75.7% accuracy rate with just six features [12]. We introduced a model for 100-hidden-neurons, four-layer deep neural networks. Adam, an adaptive moment estimation method, is used as an optimizer for deep neural networks (DNNs) in this study. By looking at many smaller parts of the KDD99 dataset, the writers were able to get very good results. The original NSL-KDD dataset was reorganized and revalidated. This paper offers a model for Intrusion Detection System (IDS) prediction based on a Stacked Sparse Auto-encoder (SSAE), which achieves up to 98% accuracy. in [13] You can get better precision by using DNN for IDS along with tenfold cross-validation on the original NSL-KDD dataset.

For Intrusion Detection Systems (IDS), Yin et al. suggested a Recurrent Neural Network (RNN)-based network. Binary and multi-classification approaches in this research achieved accuracies of 83.3% and 81.3%, correspondingly. [14] Used SNDAE, or Stacked Non-symmetric Deep Autoencoder, to identify cyberattacks on networks. On the KDDcup99 dataset, the suggested method reached a precision of 97.85%, and on the NSL-KDD dataset, it reached a precision of 85.42%. Also, for IDSs, we recommend models that use Auto-Encoders (AEs) and Long Short-Term Memory (LSTM). This study also compares and contrasts regular machine learning models with the suggested method when used on the NSL-KDD dataset.

[16] The model's architecture had both CNN and LSTM layers, which allowed for the incorporation of LSTM. The suggested model's performance will be evaluated in relation to the BGRU's. The suggested model beats alternative classification methods, including MLP and LSTM-only models, according to the paper's authors. [17] A hybrid framework for intrusion detection is proposed, which integrates deep Autoencoders (AE) with Long Short-Term Memory (LSTM) networks and bidirectional LSTM (Bi-LSTM) networks. The framework initially extracts optimal features using AE, and LSTMs are then employed to classify samples as normal or anomalous. The deep learning models used in the above methods are not good for making intruder detection systems work well because they need a lot of computing power and training time. [18] in Built a 99.9% accurate binary classification model for anomaly detection using the KDDcup99 dataset. Nevertheless, this model requires more processing time due to its resource-intensive nature and the fact that it does not differentiate attacks. In addition, [19] used hybrid SVMs to create a binary predictor that was 95.75% accurate on the KDDcup99 dataset.

Another problem is that existing IDEs aren't good at classifying data optimally. Using a special IDS architecture on the edge server, you can accurately guess different intrusions and strange behaviour 99.79% of the time. [20]. As a benchmark with different properties and classes, the proposed study utilizes the KDDcup99 dataset. This study looks at both data processing to improve ML model performance and the effects of class imbalance in the dataset [21]. This research uses feature selection to get rid of unnecessary and duplicate data from the dataset, since not all features are equally important. While researchers have used a variety of ML and DL techniques to identify intrusions in IoT networks, they have not been successful in predicting the best values for hyperparameters using only human calculations [22–29]. To solve this problem, Automated Machine Learning (Auto-ML) applies multiple ML algorithms to the dataset without human intervention, optimizing the development procedure and classification accuracy for each technique and hyperparameter. After the learner finds another optimal value [36–39], it continues to use the results of the previous classifier to choose different techniques [30–35] and hyperparameters**.**

PROPOSED METHODOLOGY

A model framework is constructed in order to categorize network traffic as either routine traffic or a particular type of assault. There are examples of network traffic that concurrently express the value or status of each characteristic, as well as a number of features that have been received from a range of sources. These examples of network traffic can be found in the following sentence. OSRI is utilized in the proposed technique in order to address the issues of under-sampling rich classes and over-sampling redundant classes during the sampling process. By utilizing these preprocessing procedures, the examples of all classes contained within the dataset are brought into equilibrium. This provides the classifier with the ability to comprehend the patterns that are concealed within the instances. Through the utilization of the Mutual Information (MI) index, a selection of characteristics is made at this time.

After that, these features are classified with the assistance of supervised learning, which is a method of Ensemble Machine Learning. The classification of these features is then compared to the classification of other algorithms. MI helps to identify all of the aspects that are either completely irrelevant or have a small influence on classification, which enables them to be removed from consideration. Calculating the weightage of each feature in the categorization of class labels is the method that is utilized to get this conclusion.

Figure 1 is a visual representation and summary of the process that is related with the paradigm that was suggested. Using fivefold cross-validation on training data is one way to eliminate the risk of overfitting. This is accomplished by utilizing the data. A total of eighty percent of the dataset is utilized for the goal of training, while the remaining twenty percent of the data is utilized for the purpose of testing the model. Following that, the strategy that was suggested would be implemented on a server that is situated on the periphery in order to identify any threats or irregularities that may have occurred. The algorithm is installed on edge servers in order to do traffic analysis on the network. This is due to the fact that edge servers have a quicker response time than cloud servers. A lightweight machine learning method is utilized for the purpose of data classification. This is due to the fact that it takes less processing power and storage space than Deep Learning methods. The information is classified as a result of this reason.

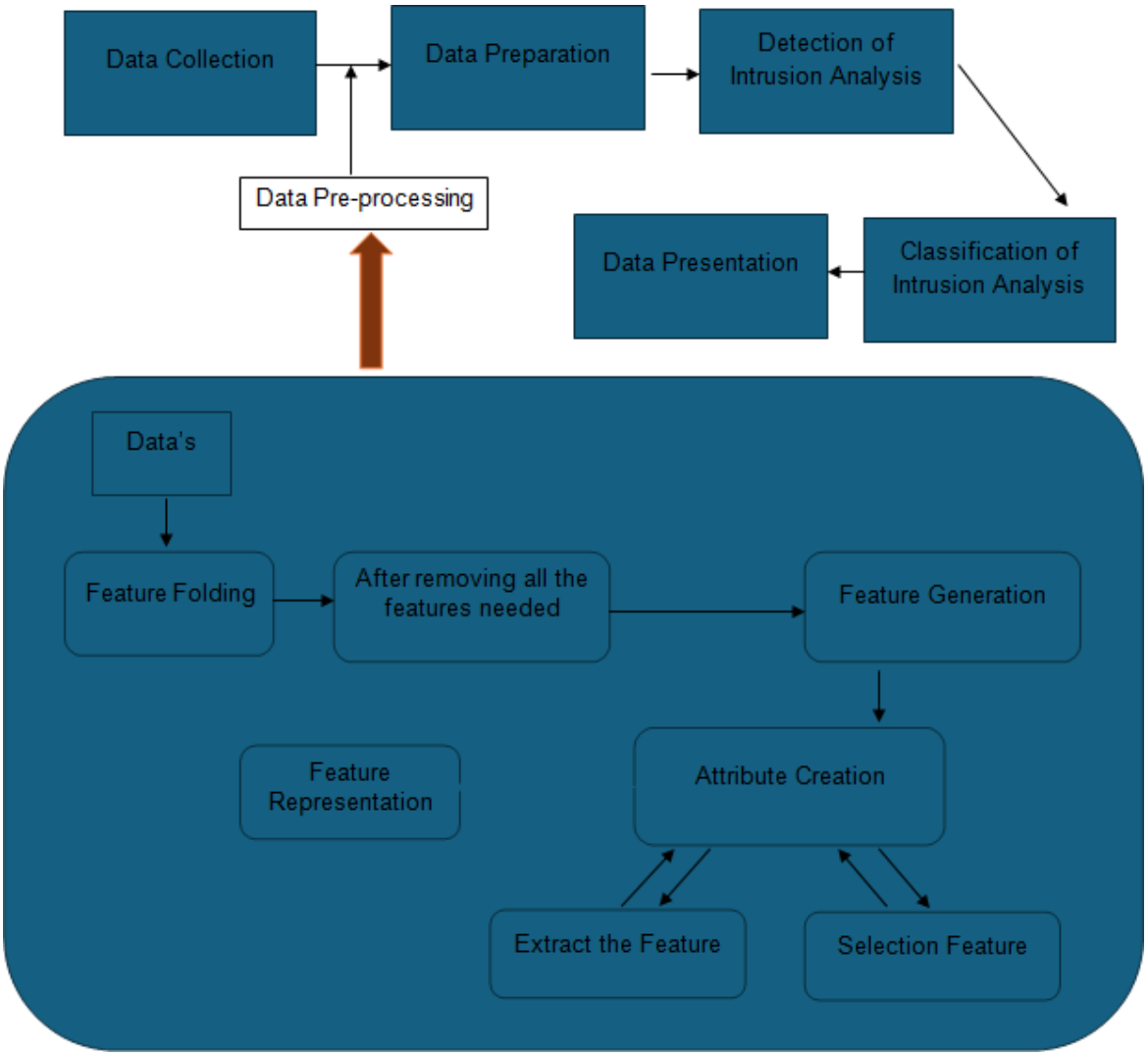


Figure 1. ML based Feature Classification Network Model

1. Feature extraction for differentiating several languages using statistical parameters and a rule-based methodology: Extraction of features from binary bit streams using statistical parameters through the conversion of binary bit streams into real-valued sequences (BTR). The resultant real-valued sequence will undergo statistical analysis to extract its characteristics. The statistical methods utilize

probability and insights obtained automatically from a language-representative text corpus. Feature extraction via a rule-based approach.

2. Normalization of Attributes: Euclidean distance is the primary measure used to determine the closeness between two patterns in a pattern-based learning framework.
3. Choosing features: Choosing features is very important because picking features with low discriminatory power will result in poor classifier performance. When developing classifiers, however, information-rich features would yield better outcomes. Selecting features that improve between-class distance and decrease within-class variation in the feature vector space should be our goal.
4. Using machine learning techniques, including MDC, PLC, MLC, and LSC, to achieve exceptional performance in classifying encrypted and plain bit streams across multiple languages.
5. Functional approximation is a great way to optimize the visual representation of high-dimensional data. Based on high-dimensional data, the functional approximation method has created a two-dimensional graphical representation. People can talk about the transformation that will make one variable the dependent variable in higher-dimensional data. Once plotted against the variable, this function provides a two-dimensional functional representation of the initial data.

Figure 1 illustrates the workflow of the proposed model and provides a description of it. Employing fivefold cross-validation on training data mitigates the risk of overfitting. The proposed technique will be implemented on a peripheral server to detect abnormalities or attacks. Edge servers exhibit decreased latency compared to cloud servers, since they necessitate reduced storage and processing power, rendering them optimal for controlling waiters in Algorithm 1.

Algorithm 1: Feature Generation Algorithm

{

Input:NSL-KDD Dataset {OSRI}

Output: Feature Subset Vector

Step-1: First, the records are loaded and analyzed, and the IDS dataset is considered. To extract features, the records are thoroughly examined. Loading and analyzing the dataset is done as

$$\text{Ext\_Fea(set(i))} = \sum_{i=1}^{M} \frac{\text{getfeat (set (i))}}{\text{count (set)}} + \text{getvalue(set(i)} \in \text{ADDS} \quad (1)$$

Step-2: Following loading, the dataset will analyze each record by pulling out all of its features. Using the dataset as a basis, the feature extraction process extracts every feature as

$$\text{Feat\_(Ext(OSRI))} = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\text{get\_attr (set (i))}}{\text{count (Ext (i))}} - \sum_{i=1}^{M} \max(\text{get\_value(j))} - \min(\text{get\_value(j))} + \text{Th} \quad (2)$$

Th→ Threshold value is utilized as a fixed value by the feature extraction model.

Step-3: We can identify the most linked aspects for recognition by prioritizing the gathered information. The following is a ranking of the extracted features.

$$P\_FE(Feat\_(i)) = R' + \delta\left(\sum_{i=1}^{M} \max(Feat\_(Ext(i))) + \frac{\lambda}{\min\_(Feat\_(R(i))}\right) - Th \quad (3)$$

In this situation, the feature correlation value→R', is derived by comparing the feature values. $R(i)$ →Function that is used to retrieve the relevant feature vector from the feature values. The number of features extracted →λ.

Step-4: Based on the allocated priorities, the feature vector is generated that is used for accurate anomaly detection. The feature vector generation is performed as

$$FVset(\Pr i(i)) = \max(\Pr i(Feat(i)) + \left(\Pr i(\text{Re}\, c(i)) + \sum_{i=1}^{M} G' - \min(Feat(i))\right) \quad (4)$$

G' → Groups most prioritized features for disease detection.

Step-5: An input random vector is utilized to create a sample inside the scope of the generator model. A random vector produced from a Gaussian distribution serves as the seed for the producing process. Locations within this multidimensional feature space will be associated with the domain using this compressed form of the data distribution. The distribution is carried out as

$$P' = \max_{\text{Re}\, c(i) \in OSRI} FV\_set_i^M + \sum_{i=1}^{N} diff(Fea\_sub[V]\_set(\max(\Pr i(i), \Pr i(i+1)))) \quad (5)$$

Step-6: The final feature subset vector creation is based on the Gaussian distribution outputs. Ultimately, a feature subset vector is produced as

$$Fea\_sub[V\_set'] = \left(Max(P') - \delta\right)/\lambda \quad (6)$$

$$Fea\_subV(\text{Pri}') = \frac{\sum_{i=1} \exp\left(\max(Fea\_Extset(i)) + Th - G'\right)}{count(Fea\_Extset(i))} + R' \quad (7)$$

In order to implement the specified features, an optimization technique must first be devised. This algorithm is what makes whale behaviour as smart as spindle cells, as shown in Figure 2. Its network-like communication capabilities allow it not just great manoeuvrability, but also the ability to slow down or even reverse course. To build a cluster or group, the meta heuristics approach is used, which entails using previous data sets. Whenever new data is added to the network, it will be allocated to one of the clusters. The cluster that is assigned to a new set of changes is then modified once more. Whale optimization is represented by the letter 'W' and can vary from 1 to N. Because it is two-dimensional and has two clusters, the centroid of this perfect solution, W(N), may be located. This helps with the cluster's creation, as seen in algorithm 2.

$W(N) = \{Clas_1;\ Clas_2\};$ (1)

Centroid point of cluster can be determined based on the elements by the request as represented Cent (N).

$Cent(N) = \{Req\ [IData\ (1) \ldots\ldots IData(N)];\ IData\ [Ele\ (1) \ldots\ldots IData(N)];$ (2)

Algorithm 2:Modified Whale optimization algorithm

Input: Number of Whales consider W(N)

Output: Fitness Function

// **Cluster formation**

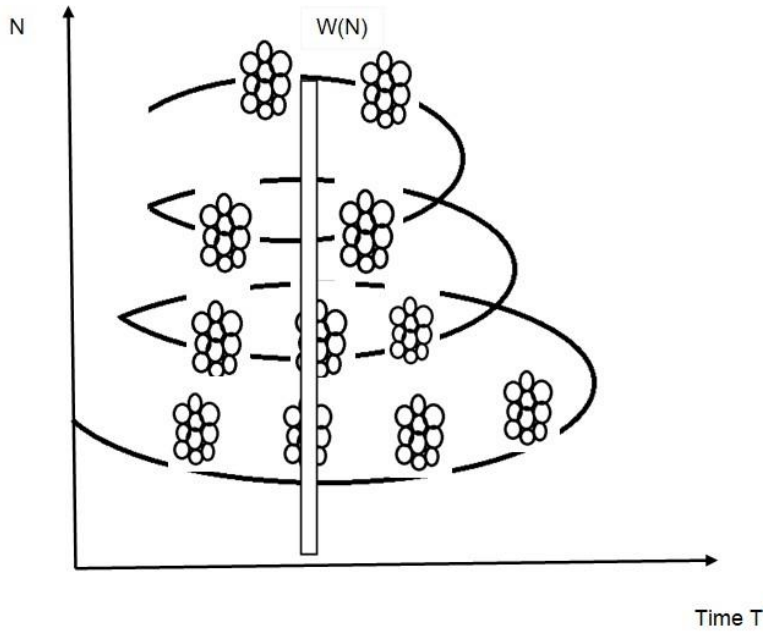
1. Set the starting value of the whale count, W(N)
2. The centroid, Cent(N), is calculated using Eqn, (2), using the constituents of the desired node.
3. Find out where the data is in relation to the cluster.

// **Fitness Function**

To determine the fitness function for each node represented as Whale

1. W(N) is the optimal point on the graph.
2. Assuming that during iteration (max) is true,
3. Choosing a whale as a node: 3.
4. Establish the fitness function. (W(N)) Fit
5. Assuming the Best node is used, the Fitness function Min(Fit(W(N)))
6. W(N) for every element
7. Make the necessary changes to the values of Req(Ele(N)), Cent(N), and p; \
8. Use Equation (3) to generate a random location vector.
9. Otherwise
10. Bring the node's (Whale's) location up to date.
11. Revised fitness assessment
12. One plus T equals T.
13. Final Output
14. Update on W(N)

$ssW(T + 1) = Rand(N) - A.C$ (3)



ss

Figure 2. Whale based spindle representation

Performance Analysis

A multi-class classifier is modeled to forecast various assaults and to determine when system circulation is regular. The NSL-KDD dataset, which is extensively used but has not received much attention for its ability to categorize different abnormalities and assaults, is utilized to train the model. The classification of classes can reduce harm and eliminate the threat quickly. The results are created using MATLAB on a PC with 16 GB of RAM and an Intel Core i5 processor.With MATLAB's classification learner program is adjusted.

For example, to get rid of the chance that uneven data will give wrong results. Conversely, the "smurf" class, which makes up more than 50% of the sample, is subject to under-sampling. The classification results on raw data and modified data also demonstrate that the testing accuracy improved significantly for the latter, due to the model being able to learn the patterns better and modify themselves accordingly. Table 2 shows the mutual information index of each feature. The last two features shown, Inum_outbound_cmds and is_host_login, don't have any effect on the classifier, and the three features next to them. It demonstrates that attributes such as priority, outgoing command, and sender and recipient's byte count do not help determine whether an anomaly or attack occurs. Certain features are removed to reduce the dataset's size and processing time.

Table 1: Classification Accuracy for Raw Data

	Training Accuracy	**Testing Accuracy**
Models		
KNN	99.70%	94.60%
SVM	99.67%	94.57%
NN	98.50%	94.43%
Ensemble	99.93%	88.73%
FGen-MWO	99.98%	97.98%

Table 2: Classification for Processed Data

	Training Accuracy	**Testing Accuracy**
Models		
KNN	99.60%	98.50%
SVM	99.87%	94.57%
NN	98.70%	94.43%
Ensemble	99.95%	88.73%

FGen-MWO	99.99%	97.99%

CONCLUSION

A unique framework that can distinguish between 22 different forms of anomalous behaviours is designed to categorize data from IoT networks as malicious or usual. An analysis of the outcomes produced from the balanced and unbalanced datasets reveals that while oversampling and undersampling are employed to achieve balance, the model's performance in the testing instances is subpar compared to the raw dataset.Feature Generation with Modified Whale Optimization (FGen-MWO) Algorithm to equilibrate the NSL-KDD dataset. This reduces bias in machine learning models, hence enhancing their overall performance. MI-based attribute collection involves selecting attributes that are more pertinent to the output class. MI evaluates the correlation of each feature in the dataset with the final class and eliminates features with low MI scores. Consequently, the dataset is reduced, thereby decreasing the training costs and durations of the machine learning modelss. The suggested model not only achieves the highest accuracy using a benchmark dataset, but it is also lightweight and takes less time to test.

As the software used in end devices evolves rapidly, as does the advent of cutting-edge technologies in IoT networks, so are the sorts of assaults. Researchers might consider developing an intelligent program for future projects. It could incorporate transfer learning to incorporate newly identified attack types into the training dataset, facilitating subsequent testing. Reinforcement learning could help achieve this goal by classifying the data in case of a new attack.

RFERENCES

[1] Abdoh SF, Abo Rizka M, Maghraby FA (2018) Cervical cancer diagnosis using random forest classifier with SMOTE and feature reduction techniques. IEEE Access 6:59475–59485.

[2] Akashdeep S, Manzoor I, Kumar N (2017) A feature reduced intrusion detection system using ANN classifier. Expert SystAppl 88:249–257. https://doi.org/10.1016/J.ESWA.2017.07.005

[3] Albulayhi K, Smadi AA, Sheldon FT, Abercrombie RK (2021) IoT intrusion detection taxonomy, reference architecture, and analyses. Sensors 21(19):6432.

[4] Aslam MS, Tiwari P, Pandey HM, Band SS, El Sayed H (2023) A delayed Takagi-Sugeno fuzzy control approach with uncertain measurements using an extended sliding mode observer. InfSci 643:119204

[5] Aslam MS, Tiwari P, Pandey HM, Band SS (2022) Observer-based control for a new stochastic maximum power point tracking for photovoltaic systems with networked control system. IEEE Trans Fuzzy Syst

[6] Bano S, Hussain SF (2022) Prediction of Covid-19 and post Covid-19 patients with reduced feature extraction using Machine Learning Techniques, pp 37–42. https://doi.org/10.1109/FIT53504.2021.

[7] Bilal H, Yin B, Kumar A, Ali M, Zhang J, Yao J (2023) Jerk-bounded trajectory planning for rotary flexible joint manipulator: an experimental approach. Soft Comput 27(7):4029–4039.

[8] Bovenzi G, Aceto G, Ciuonzo D, Persico V, Pescape A (2020) A hierarchical hybrid intrusion detection approach in IoT scenarios. In: 2020 IEEE global communications conference GLOBECOM 2020—proceedings, vol 2020-January.

[9] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J ArtifIntell Res 16:321–357.

[10] Entropy, Relative Entropy and Mutual Information Eskandari M, Janjua ZH, Vecchio M, Antonelli F (2020) Passban IDS: an intelligent anomaly-based intrusion detection system for IoT edge devices. IEEE Internet Things J 7(8):6882–6897.

[11] Gao Y, Liu Y, Jin Y, Chen J, Wu H (2018) A novel semi-supervised learning approach for network intrusion detection on cloudbased robotic system. IEEE Access 6:50927–50938.

[12] Gill SS, Garraghan P, Buyya R (2019) ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices. J SystSoftw 154:125–138.

[13] Goel S, Williams K, Dincelli E (2017) Got phished? Internet security and human vulnerability. J AssocInfSyst 18(1):2.

[14] Hsu CM, Azhari MZ, Hsieh HY, Prakosa SW, Leu JS (2021) Robust network intrusion detection scheme using long-short term memory based convolutional neural networks. Mob NetwAppl 26(3):1137–1144.

[15] Hussain SF, Ashraf MM (2023) A novel one-vs-rest consensus learning method for crash severity prediction. Expert SystAppl 228:120443.

[16] Khan FA, Gumaei A, Derhab A, Hussain A (2019) TSDL: a two-stagedeep learning model for efficient network intrusion detection.IEEE Access 7:30373–30385.
[17] Kim J, Shin N, Jo SY, Kim SH (2017) Method of intrusion detectionusing deep neural network. In: 2017 IEEE international conferenceon Big Data and smart computing, BigComp2017,pp 313–316.
[18] Kraskov A, Sto¨gbauer H, Grassberger P (2004) Estimating mutualinformation. Phys Rev E Stat Phys Plasmas Fluids RelatInterdiscip Top 69(6):16.
[19] Manavalan E, Jayakrishna K (2019) A review of Internet of Things(IoT) embedded sustainable supply chain for industry 4.0requirements. ComputIndEng 127:925–953.
[20] Mushtaq E, Zameer A, Umer M, Abbasi AA (2022) A two-stageintrusion detection system with auto-encoder and LSTMs. ApplSoftComput 121:108768.
[21] Nahavandi D, Alizadehsani R, Khosravi A, Acharya UR (2022)Application of artificial intelligence in wearable devices:opportunities and challenges. Comput Methods ProgramsBiomed 213:106541.
[22] Nimbalkar P, Kshirsagar D (2021) Feature selection for intrusiondetection system in Internet-of-Things (IoT). ICT Express7(2):177–181.
[23] Raghuvanshi A, Singh UK, Joshi C (2022) A review of varioussecurity and privacy innovations for IoT applications in healthcare.In: Advanced healthcare systems empowering physicians14480
[24] Sicato JCS, Singh SK, Rathore S, Park JH (2020) A comprehensiveanalyses of intrusion detection system for IoT environment. J InfProcess Syst 16(4):975–990.
[25] Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M (2016)Deep learning approach for Network Intrusion Detection inSoftware Defined Networking. In: Proceedings - 2016 internationalconference on wireless networks and mobile communications,WINCOM 2016: green communications and networking,2016, pp 258–263.
[26] Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailedanalysis of the KDD CUP 99 data set. In: IEEE symposium oncomputational intelligence for security and defense applicationsCISDA 2009.
[27] Wang L, Q. Zhai Q, Yin B et al (2019) Second-order convolutionalnetwork for crowd counting. In: Proceedings of the SPIE 11198,fourth international workshop on pattern recognition, 111980T.
[28] Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, VapnikV(2000) Feature Selection for SVMs. Adv Neural Inf Process Syst13:2000
[29] Wu Y, Dai HN, Wang H, Xiong Z, Guo S (2022) A survey ofintelligent network slicing management for industrial IoT:integrated approaches for smart transportation, smart energy,and smart factory. IEEE CommunSurv Tutor 24(2):1175–1211.
[30] Ramya Manikyam, J. Todd McDonald, William R. Mahoney, Todd R. Andel, and Samuel H. Russ. 2016.Comparing the effectiveness of commercial obfuscators against MATE attacks. In Proceedings of the 6th Workshop on Software Security, Protection, and Reverse Engineering (SSPREW'16)
[31] R. Manikyam. 2019.Program protection using software based hardware abstraction.Ph.D. Dissertation.University of South Alabama.
[32] P Kanagala, R Jayaraman, FAA-Cloud approach to minimize computation overhead using fuzzy-based crypto security, Soft Computing, 1-11.
[33] P Kanagala, R Jayaraman, Effective encryption approach to improving the secure cloud framework through fuzzy-based encrypted cryptography, Soft Computing, 1-10.
[34] R Pulimamidi, P Ravichandran, Connected Health: Revolutionizing Patient Care Through Artificial Intelligence Innovations, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 3940-3947.
[35] R Pulimamidi, P Ravichandran, Enhancing Healthcare Delivery: AI Applications In Remote Patient Monitoring, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 3948-3954.
[36] Naga Simhadri Apparao Polireddi, K Chaitanya, Web accessibility evaluation of private and government websites for people with disabilities through fuzzy classifier in the USA, Soft Computing, Pages 1-9. 2023.
[37] Adithya Padthe Srinivas Aditya Vaddadi , Pandu Ranga Rao Arnepalli , RamyaThatikonda, , Effective Malware Detection Approach based on Deep Learning in Cyber-Physical Systems, International Journal of Computer Science and Information Technology, Volume 14, Issue 6, Pp 01-12.
[38] Mythili, R., bama, B. S., Kumar, P. S., Das, S., Thatikonda, R., &Inthiyaz, S. (2023). Radial basis function networks with Lightweight Multiscale Fusion strategy-based underwater image enhancement. Expert Systems. https://doi.org/10.1111/exsy.13373, (Publisher: Wiley)
[39] Vaddadi, S. A., Thatikonda, R., Padthe, A., &Arnepalli, P. R. (2023). Shift-Left Testing Paradigm Process Implementation for Quality of Software Based on Fuzzy. https://doi.org/10.21203/rs.3.rs-2845536/v1, (Publisher: Springer Link)
[40] sss