EAI Endorsed Transactions
On Scalable Information Systems
Research Article: EAI.EU

Knox: Lightweight Machine Learning Approaches for Automated Detection of Botnet Attacks

Shritik Raj[1,*], Bernard Ngangbam[2], Sanket Mishra[3], Vivek Gopalasetti[4], Ayushi Bajpai[5], Ch.Venkata RamiReddy[6]

[1,2,3,4,5,6]School of Computer Science & Engineering, VIT-AP University, Amaravati, 522237, Andhra Pradesh, India

Abstract

With an advancement in technology, the Internet of Things (IoT) has penetrated various domains such as smart buildings, intelligent transportation systems, healthcare, smart parking, air quality monitoring, water contamination identification, and supply chain owing to its ubiquitous nature. IoT devices periodically collect the data and send it to the gateway or server for pre-processing. However, the security offered in the IoT devices or gateways are still in a nascent stage. An Intrusion Detection System (IDS) meant for detecting the cyber threats on IoT should intercept most threats with minimum latency and yet be lightweight in nature. IoT devices also have low memory footprint which makes them resource constrained. This paper presents a framework built using a three-tier IoT architecture that successfully detects most attacks using machine learning approaches with an accuracy of 99%. Machine learning approaches are fed data using Apache Kafka to REST API. Sampling methods such as undersampling and adaptive synthetic sampling are applied to balance the imbalanced nature of the dataset. We examined the robustness of the approach using different samples with varying sizes and varying dimensions. Experimental results depict a superior performance of random forest over other approaches in terms of speed and accuracy.

Received on 14 June 2023; accepted on 26 August 2023; published on 26 September 2023

Keywords: Machine Learning, Botnet Detection, Internet of Things, Dimensionality Reduction, Data Sampling Techniques, Data streaming, Feature Extraction

doi:10.4108/eetsis.3997

1. Introduction

In today's world of technology and artificial intelligence, the Internet of Things (IoT) devices are very much relevant to daily usage in terms of security, analysis, data collection and a plethora of technological developments in the past decade. With the rapid growth in the field of artificial intelligence with newer technologies such as deep learning technologies, IoT, big data analytics, cloud computing, real-time streaming platforms, the need of the security aspect of these technologies need to be safeguarded properly as we are talking here about a vast ocean of data. IoT basically is the root of all smart systems as it deals with sensors and data using artificial intelligence and data analytics technologies for detecting outliers, speech translation, biometric systems, health-care smart systems, smart homes with security, soil quality monitoring smart agriculture, smart sensors for data collection in the retail industry, and so on.

In the recent past the growth of IoT devices has risen exponentially, with a forecast of being almost double from 50.1 billion in 2020 to more than 100 billion devices by 2030. All major industries are connected with almost more than 100 million IoT devices working in the front-line of their architecture. Industries such as health care, retail, agriculture, finance, manufacturing, energy, hospitality, water pollution, smart homes, transportation and logistics, and so on are some which have gained fruit from the use of IoT technology. With the explosive growth of these devices, the security and risk prone to malicious attacks factor comes hand in hand with this exponential rise.

Fig 1. depicts the exponential growth of IoT devices showing an explosive rise and a forecast of an even steeper slope in the near future. IoT devices are prone

*Corresponding author. Email: rajshritik03@gmail.com

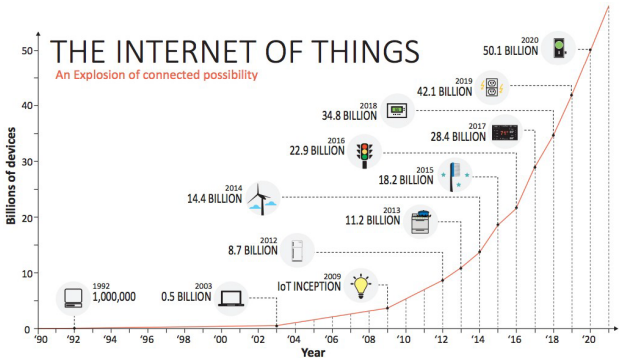

Figure 1. Growth rate of IoT devices

to intrusion attacks which means that the seamless information exchange between the connected physical devices is disrupted maliciously so an Intrusion Detection System (IDS) is essential. We can have different forms of attacks such as Denial of Service (DoS), Mirai, Man in the Middle (MITM), Scan attacks, namely which are some common types of attacks in IoT systems. An anomaly identification system is essential for improving the security of the network architecture of these systems. This paper is based on the creation of an IDS for automatic detection of botnet attacks using lightweight machine learning models on the IoT node itself.

2. Related Work

Detecting botnet attacks has been a challenge for cybersecurity researchers nowadays, spotting these attacks has become an active area of research in recent years. Several studies have proposed machine learning-based approaches for automated botnet detection. In this section, we provide a brief overview of some of the most relevant work in this field. Ullah and Q. H. Mahmoud et al. [1] took the IoTDI dataset, and applied Shapiro-Wilk feature ranking algorithm. They then further classified the problem into binary classification and multi-class classification, i.e., Normal and Anomaly for binary, and DoS, Mirai, Scan, and MITM for multi-class. They applied SVM, GaussianNB, LDA, Logistic Regression, Decision Tree, Random Forest and Ensemble, achieving the highest score of 87% using ensemble model.

N. Koroniotis, N. Moustafa, E. Sitnikova and B. Turnbull, et al. [2] evaluated the reliability of Bot-IoT dataset using different statistical and machine learning methods. They proposed a new datset Bot-IoT upon which they built a baseline for allowing botnet identification across IoT networks.

More recently, M. K. Yadav and K. P. Sharma et al. [3] mainly focuses on providing analytical studies of existing IDS systems and explores the ways to create an effective IDS using several machine learning algorithms. This work gave us the motivation to propose Knox for the same.

Another recent study by S. Krishnaveni, P. Vigneshwar, S. Kishore, B. Jothi and S. Sivamohan, et al. [4] proposed an effective framework for detecting anomalies in cloud computing systems using support vector machines with an increased accuracy of 96% benchmark also minimising false alarm rates. False alarm are special cases which need to be handles thus, the Knox Framework deals it using optimised retraining in case of anomalies.

In our work, we build on these previous studies and proposed a lightweight machine learning approach for automated detection of botnet attacks. We utilized the strengths of both unsupervised and supervised learning techniques and used a lightweight machine learning algorithm that can be deployed on resource-constrained devices. While previous studies have proposed similar machine learning-based approaches, this study's framework provides superior accuracy (99%) in detecting most botnet attacks. Additionally, the Knox framework's random forest algorithm offers faster and more accurate results.

3. Methods

3.1. Model Architecture

The novel architecture that we have used for creating an automated system for detection of botnet attacks proposed in this work is the Knox Framework as shown in Fig 2.

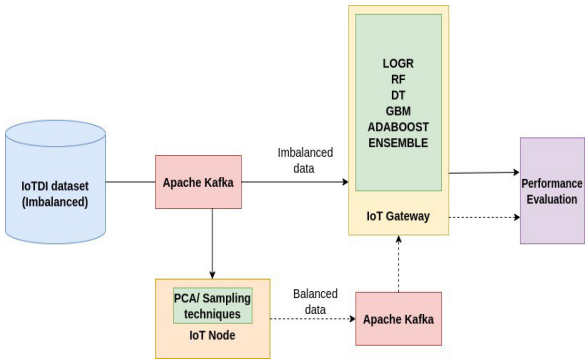

Figure 2. Knox Framework

3.2. Dataset Description and Pre-processing

We have considered the IoTDI dataset [1] for identifying botnets. The dataset consists of 86 columns and 625783 rows. The dataset consists of 53,817,338 observations with no missing values. It also provides information that identifies the type of botnets behind the attacks. We used InfluxDB for storing the dataset on a server and pulled data using an adaptive windowing approach

using Apache Kafka Stream for training the models. We pulled random 50,000 samples from the dataset on the InfluxDB localhost server using Apache Kafka as we are working on lightweight models using a recursive self-retraining algorithm to optimize accuracy, i.e., by recursively pulling a random chunk from the dataset every time until high accuracy is obtained. We sent the data to the IoT nodes where the dimensionality reduction approaches were applied which helped in minimizing the latency of the predictions.

In this paper, we have identified botnet detection on IoT devices as a binary classification problem where the proposed approach identifies whether an attack is there or not. Fig 3. that the data does not contain any missing values. We have used Min-Max normalization [5] [6] for feature scaling. Feature Scaling helps in putting the data between the range of -1 to +1 which helps in the removal of bias in data.

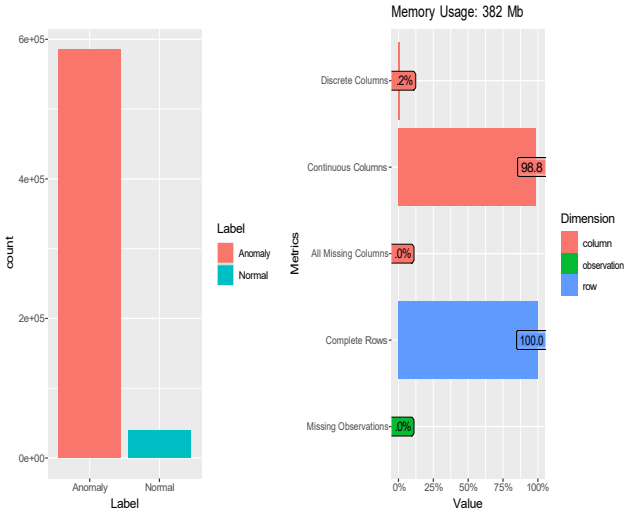

Figure 3. Class distribution and missing values of IoTDI dataset

More than 30 columns have fifty percent of values as zeroes which were dropped during the pre-processing phase. Fig 3. depicts the data distribution as data instances of anomaly were comparatively more than that belonging to normal class. The data instances belonging to the normal class and anomaly classes are 40,073 and 58,5710 respectively. The next section elaborates on the various sampling techniques used to balance the classes in the given dataset.

3.3. Data Sampling Techniques

For balancing the data, we have used various sampling techniques [7] (the data was also balanced using Synthetic Minority Oversampling Technique (SMOTE), Oversampling, and Borderline SMOTE which is not included in this work) such as, under sampling and adaptive synthetic sampling [8]. Before balancing the 50,000 random sampled data points using the under-sampling technique the number of data instances in the normal and anomaly classes were 46,339 and 3,161, respectively. Now after balancing the data using the under-sampling technique, the number of instances significantly changed to 3,161 and 3,161 in normal and anomaly classes, respectively.

3.4. Dimensionality Reduction Techniques

Feature Extraction. For feature extraction, we have used Principal Component Analysis (PCA) [9] with linear increment in the number of components parameter i.e., 2,4,6,8,10 components. The PCA was done on the IoT node level and then the new feature extracted data is then sent through the gateway for model training.

Feature Selection. Feature selection is a critical stage in the process of identifying botnets since it reduces the dimensionality of the data and gets rid of extraneous or redundant features. Here is a quick rundown of some popular feature selection methods for botnet detection information:

Pearson Correlation: The linear link between two variables is measured by the Pearson correlation [10]. This well-liked feature selection method ranks the features according to how closely they correlate with the desired variable. The features chosen are those with the highest correlation coefficients.

Chi-Squared test: The chi-squared test [11] is a statistical technique for identifying the association between two category variables. By contrasting the distribution of a characteristic with the distribution of the target variable, it is possible to assess the importance of a feature in the context of botnet detection data.

Recursive Feature Elimination (RFE): RFE [12] is a well-liked feature selection strategy that iteratively eliminates the least significant features using a machine learning model. All the features are first trained into a model, which is then ranked according to relevance. Once the desired number of features is attained, it eliminates the least significant feature and continues the procedure.

Logistics Regression Feature Selection: A logistic function is used in logistic regression, a binary classification procedure, to forecast the likelihood of the target variable. The features with the greatest coefficient values are chosen when applying logistic regression.

Random Forest Feature Selection: Several decision trees are combined using the ensemble learning method random forest to get a more precise model. When utilising random forest, feature selection entails choosing the features with the highest importance scores.

LightGBM: LightGBM [13] is a gradient boosting framework that forecasts the target variable by using

decision trees. Choosing features in LightGBM entails choosing those with the highest gain scores.

	Feature	Pearson	Chi-2	RFE	Logistics	Random Forest	LightGBM	Total
1	Src_Port	True	True	True	True	True	True	6
2	Dst_Port	True	True	True	True	True	True	6
3	SYN_Flag_Cnt	True	True	True	True	False	False	4
4	Flow_Duration	False	False	True	True	True	True	4
5	Flow_Byts/s	True	False	True	True	False	True	4
6	Protocol	False	False	True	True	False	False	2
7	Pkt_Size_Avg	False	False	True	True	False	False	2
8	Pkt_Len_Mean	False	False	True	True	False	False	2
9	Init_Bwd_Win_Byts	False	False	False	False	True	True	2
10	Idle_Min	True	False	False	False	False	True	2
11	Idle_Max	False	False	False	False	True	True	2
12	Flow_Pkts/s	False	False	False	False	True	True	2
13	Flow_IAT_Max	False	False	False	False	True	True	2
14	Bwd_Pkts/s	False	False	False	False	True	True	2
15	Bwd_Header_Len	False	False	True	True	False	False	2
16	Pkt_Len_Var	False	False	True	False	False	False	1
17	Pkt_Len_Max	False	False	False	True	False	False	1
18	Idle_Mean	True	False	False	False	False	False	1
19	Fwd_Seg_Size_Avg	False	True	False	False	False	False	1
20	Fwd_Pkt_Len_Std	False	True	False	False	False	False	1

Figure 4. Feature selection results for all features by using different approaches

3.5. Data Streaming

Now, that we have the feature selected data ready for training, we pulled the data from Influx DB localhost server into an Apache Kafka broker, using Kafka Connect, influx-source-connector. This data is then directly passed into the IoT node level for machine learning training, with the algorithms mentioned in section (4).

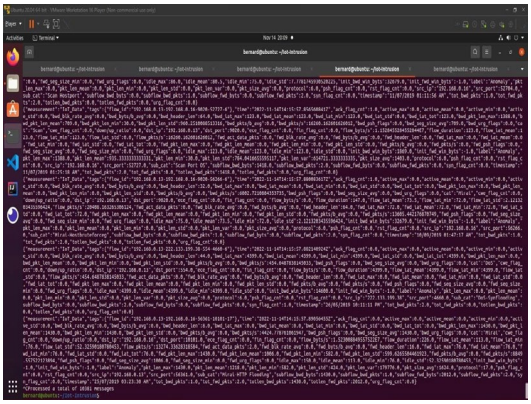

Figure 5. Data streaming from InfluxDB into the Kafka then consuming through Kafka Consumer

4. Machine Learning Approaches

Now, as we have the dimensionality reduced features using PCA passed into the IoT node level, we go for machine learning modelling using several different machine learning algorithms. Some of the used machine learning algorithms are:

4.1. Logistic Regression

The StandardScaler and LogisticRegression [?] are combined to build a pipeline. For logistic regression, StandardScaler scales the data to have a mean of 0 and variance of 1, which is crucial. The classifier divides the data into botnet and non-botnet categories using logistic regression. A set of C values and different regularisation penalty types are defined, along with a grid of hyperparameters to search. Smaller values of the C parameter, which is the inverse of regularisation strength, denote stronger regularisation. The norm employed in the penalization is specified by the penalty parameter. The best hyperparameters are then found using 5-fold cross-validation and GridSearchCV. The hyperparameter combinations in the param grid are exhaustively tested by the grid search algorithm, which then chooses the combination with the greatest score. The score represents the 5-fold cross-mean validation's accuracy.

The top hyperparameters and the associated score are then printed. The logistic regression model on the botnet dataset can then be trained and tested using these hyperparameters.

4.2. Decision Tree

It is a tree-like model where each leaf node represents a class label or a numerical value, each internal node represents a test on an attribute, and each branch reflects the test's result. Based on criteria like information gain or Gini impurity, the decision tree algorithm chooses the appropriate characteristic to split the data at each node and then recursively constructs the tree from the training data. To ensure that the class labels inside each subset are as pure as feasible, the data must be divided into homogenous subsets. We build a DecisionTreeClassifier [15] object and use a dictionary to specify the hyperparameters that will be tweaked.The decision tree classifier [16] instance and the hyperparameters that need to be tweaked are then passed as arguments to a newly created instance of GridSearchCV. We also state how many folds would be utilised for cross-validation. GridSearchCV, which tries all feasible combinations of hyperparameters and chooses the best model based on cross-validation performance, is used to train the model using the training data. The best model chosen by GridSearchCV is then used to make predictions on the test data,

and its performance is assessed using the accuracy score function. Lastly, we print GridSearchCV's top choice hyperparameters. This enables us to learn which hyperparameters are most effective for the detecting botnet attacks.

4.3. Random Forest

An extension of Decision Tree approach called Random Forest [17] combines different decision trees to increase the model's accuracy and robustness. A random collection of features and a random subset of training data are used to build each decision tree. This promotes variation among the trees and lessens overfitting. By combining all the forest's forecasts, the conclusion is reached. The dataset is first loaded and divided into the target variable (X) and features (X) (y). Then, a pipeline is built using RandomForestClassifier and StandardScaler. For random forest, StandardScaler scales the data to have a mean of 0 and variance of 1, which is crucial. To categorise the data as anomalous or not, we will utilize the RandomForestClassifier classifier.

4.4. AdaBoost

A common boosting algorithm approach called AdaBoost (Adaptive Boosting) [18] combines several weak classifiers to create a strong classifier. Using AdaBoost to detect botnets can be advantageous because it can increase classifier accuracy and handle unbalanced datasets. This is because botnets are built to avoid detection and resemble legitimate activity, detecting them can be difficult. Due to this, the dataset utilised for botnet identification is frequently unbalanced, with far less botnet traffic than non-botnet traffic. Thus, it might be challenging for a classifier to precisely identify botnet traffic. By providing the instances that were incorrectly classified greater weight during training, AdaBoost can help solve this problem by directing the classifier's attention towards the instances that were challenging to categorize. This can lead to a more accurate model, especially when dealing with imbalanced datasets.

4.5. Ensemble

Since botnet attacks frequently use cunning and complex evasion techniques, they might be difficult to spot. As they incorporate the predictions of numerous models to increase the overall accuracy and durability of the detection system, ensemble models [19] can be helpful for identifying botnet attacks. Many types of models, such as decision trees, random forests, and neural networks, can be included in ensemble models [20]. While each model in the ensemble may have unique advantages and disadvantages, by pooling their predictions, the ensemble can outperform any one model. Ensemble models can also aid in resolving the imbalanced data problem that frequently arises in botnet identification. Botnet traffic is frequently a minority class, making it difficult for a single classifier to reliably identify it. The ensemble model, on the other hand, can offer a more balanced method of detecting both botnet and non-botnet traffic by merging numerous classifiers.

Moreover, ensemble models can increase the system's robustness for detecting botnets. Botnet attacks can be extremely dynamic and constantly changing, so an ensemble model that is trained on a variety of models with 8 various traits may be more resistant to changes in the data and botnet attack tactics.

Upon training using these several algorithms, we can achieve almost similar accuracy scores in many cases. However, the point to choose the most optimal algorithm lies in choosing the most optimal algorithm in terms of time complexity, i.e., the time taken to train the model. In this research as expected generally, we get high accuracy from ensemble models and boosting algorithms, but they take comparatively higher amount of time to train the model. In the below section (5). we describe the most optimal algorithm and its parameters.

5. Experimental Results and Discussion

In this experimental study, our results show that we get the most optimal approach by undersampling data balancing technique, feature extraction with PCA(k=6) taking 2.57 seconds to train the lightweight model. It gives an accuracy score of 99%, f-1 score of 0.99 for class Anomaly, and f-1 score of 0.92 for class Normal, and an auc score of 0.99.

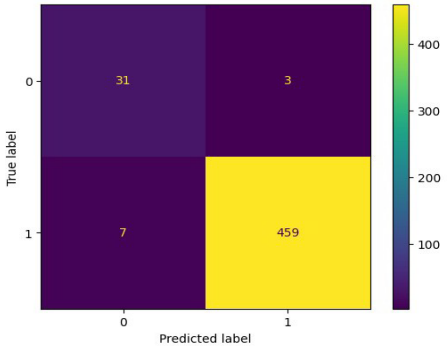

Figure 6. Confusion matrix of Random Forest model for under-sampled data with PCA(k=6)

From the above figure, we can see that we have 3 false positives(FPs) and 7 false negatives(FNs). So, we conclude that 10 datapoints are misclassified in the test set.

As, we can see in Fig 8. we get high results from the ensemble model as expected, however at PCA(k=6)

Table 1. Accuracy Scores on different PCA number of components (k) values

ML Models	PCA k values				
	k=2	*k=4*	*k=6*	*k=8*	*k=10*
Log. Regression	0.63	0.59	0.74	0.83	0.87
Random Forest	0.97	0.98	0.99	0.99	0.99
Decision Tree	0.96	0.97	0.98	0.97	0.98
AdaBoost	0.87	0.91	0.93	0.91	0.92
Ensemble	0.97	0.96	0.98	0.98	0.98

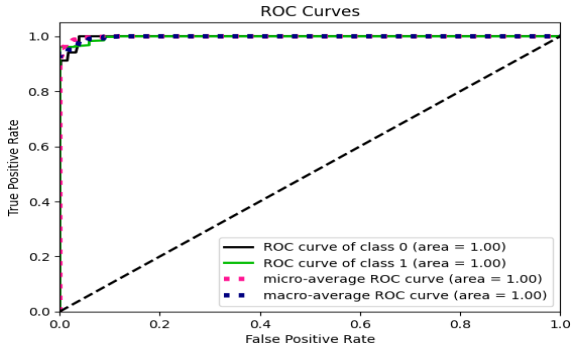

Figure 7. ROC curves of TP rates vs FP rates for PCA(k=6) Random Forest

the ensemble model took 2730.20 seconds for training, which is not optimal. It was observed during the experiment that increasing the k parameter for PCA gave slight increase in the model's accuracy for the undersampled balanced data. However, this slight increase came with a cost of increase in model training time, thus after training models for k values (2,4,6,8,10) it was observed that PCA(k=6) is the most optimal in terms of model training time using our novel Knox Framework and also in terms of accuracy. So, our most optimal algorithm by far is Random Forest with PCA(k=6) taking only 2.57 seconds.

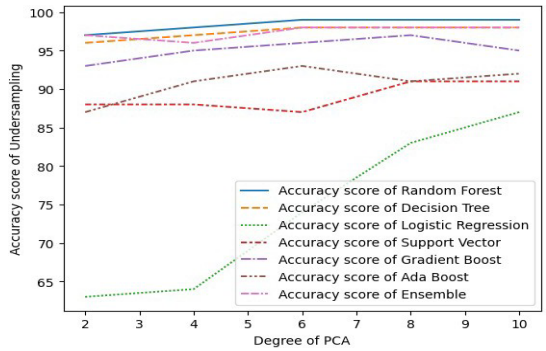

Figure 8. Accuracy score of Under-sampling Vs Degree of PCA(k)

6. Conclusion and Future Work

In this research, we used several techniques for data sampling and followed under-sampling for the final model training as it gave optimised time complexity. For faster data flowing, we used Apache Kafka locally to pull and push data into the IoT node through the gateway where the model was trained. Then, we used several feature selection and feature extraction techniques. Again, finally going with PCA approach taking an incremental number of components i.e., k = 2,4,6,8,10. From the above seen results, PCA(k=6) gave the most accurate and optimal solution in terms of time and accuracy score. From this research, the lightweight model can be run on the IoT node itself for automatic detection of botnet attacks.

For future work, we aim to extend the Knox Framework[3] to be able to classify sub-categories of Anomaly class and sub-sub-categories of it, with similar optimization in terms of time and accuracy score.

References

[1] Ullah, I. and Mahmoud, Q. (2020) *A scheme for generating a dataset for anomalous activity detection in iot networks* 508–520.

[2] Koroniotis, N. and Moustafa, N. and Sitnikova, E. and Turnbull, B. (2019) *Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset* (Future Generation Computer Systems)

[3] Yadav, M.K. and Sharma, K.P.(2021) *Intrusion Detection System using Machine Learning Algorithms: A Comparative Study* 415-420

[4] Krishnaveni, S. and Vigneshwar, Palani and Kishore, S. and Jothi, B. (2020) *Anomaly-Based Intrusion Detection System Using Support Vector Machine*

[5] Tsaramirsis and Georgios and Karamitsos and Ioannisand Apostolopoulos (2016) *Smart parking: An IoT application for smart city* (2016 3rd International Conference on Computing for Sustainable Global Development)

[6] S. Gopal Krishna Patro and Kishore Kumar Sahu (2015) *Normalization: A Preprocessing Stage 328-333*

[7] Abdelkhalek and Ahmed and Mashaly and Maggie (2023) *Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning* (The Journal of Supercomputing)

[8] Koroniotis, N. and Moustafa, N. and Sitnikova, E. and Turnbull, B. (2019) *Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset*

[9] Letteri and Ivan and Antonio and Dyoub and Giuseppe (2020) *A Novel Resampling Technique for Imbalanced Dataset Optimization*

[10] Sehgal, Shruti and Singh, Harpreet and Agarwal, Mohit and Bhasker, V. (2014) *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)* (Data analysis using principal component analysis)

[11] Nasir, Inzamam and Khan, Muhammad and Yasmin, Mussarat and Shah, Jamal and Gabryel, Marcin (2020) *Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training* (Sensors)

[12] Singhal, Richa and Rana, Rakesh (2015) *Chi-square test and its application in hypothesis testing* (Journal of the Practice of Cardiovascular Sciences)

[13] Chen, Xue-wen and Jeong, Jong Cheol (2008) *Enhanced recursive feature elimination 429-435*

[14] Alkasassbeh, Mouhammd and Abbadi, Mohammad and Al-Bustanji, Ahmed (2020) *LightGBM Algorithm for Malware Detection 391-403* (Sensors)

[15] Besharati, Elham and Naderan, Marjan and Namjoo, Ehsan (2019) *LR-HIDS: Logistic Regression Host-based Intrusion Detection System for Cloud Environments* (Journal of Ambient Intelligence and Humanized Computing)

[16] M M, Savitha and Basarkod, P I (2022) *Random Forest based Intrusion Detection System for AMI* (2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAECC))

[17] Morched Derbali and Seyed Mohammed Buhari and Georgios Tsaramirsis and M. Stojmenovic and Houssem Jerbi and Mohamed Naceur Abdelkrim (2017) *Water Desalination Fault Detection Using Machine Learning Approaches: A Comparative Study* (IEEE Access)

[18] Seyedeh Mahsan Taghavinejad and Mehran Taghavinejad and Lida Shahmiri and Mohammad Hossein Zavvar (2020) *Intrusion Detection in IoT-Based Smart Grid Using Hybrid Decision Tree* (2020 6th International Conference on Web Research (ICWR))

[19] Salman Rachmadi and Satria Mandala and Dita Oktaria (2021) *Detection of DoS Attack using AdaBoost Algorithm on IoT System* (2021 International Conference on Data Science and Its Applications (ICoDSA))

[20] Ivan Cviti and Dragan Perakovi and Marko Peria (2021) *Ensemble machine learning approach for classification of IoT devices in smart home* (International Journal of Machine Learning and Cybernetics)