iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 10 **Issue:** XI **Month of publication:** November 2022

**DOI:** https://doi.org/10.22214/ijraset.2022.47365

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue XI Nov 2022- Available at www.ijraset.com*

# Using Machine Learning in Detecting IOT Cyber Attacks

Neeraj Patil
*Department of Electronics Engineering, Vivekanand Education Society's Institute of Technology*

***Abstract:*** ***The Internet of Things (IoT) combines hundreds of millions of devices that are able to interact with each other with minimal user interaction. IoT is one of the fastest growing areas of computing; however, the reality is that in the extremely hostile environment of the Internet, IoT is vulnerable to many types of cyberattacks. Practical countermeasures to secure IoT networks, such as network anomaly detection, need to be implemented to address this issue. Regardless of the fact that attacks cannot be completely avoided forever, early detection of an attack is essential for practical defense. As IoT devices have low storage capacity and low computing power, traditional high-end security solutions are not suitable for IoT system protection. IoT devices are also now connected without human intervention for longer periods of time. This means that intelligent network-based security solutions such as machine learning solutions need to be developed. Although many studies in recent years have discussed the application of Machine Learning (ML) solutions to attack detection problems, little attention has been paid to attack detection specifically in IoT networks. In this study, we aim to contribute to the literature by evaluating various machine learning algorithms that can be used to quickly and effectively detect IoT network attacks. A new Bot-IoT dataset is used to evaluate different detection algorithms. Seven different machine learning algorithms were used in the implementation phase and most of them achieved high performance. New features were extracted from the Bot-IoT dataset during implementation and compared with literature studies, and the new features provided better results.***

***Keywords: Internet of things , Cyberattacks, Machine Learning , Cyber security ,Anomaly Detection***

## I. INTRODUCTION

Security and privacy concerns regarding computer networks are growing in the world, and computer security has become a requirement due to the proliferation of information technology in everyday life. The increase in the number of Internet applications and the advent of modern technologies such as the Internet of Things (IoT) are accompanied by new and recent efforts to invade computer networks and systems. The Internet of Things (IoT) is a set of interconnected devices where devices can connect without the need for human intervention. With IoT, many things that have sensors (such as coffee machines, lights, bicycles, and many others) in areas such as healthcare, agriculture, transportation, etc. can connect to the Internet[1]. By saving time and resources, IoT applications are changing our work and lives. It also has unlimited benefits and opens up numerous opportunities for knowledge exchange, innovation and growth.

Every security threat on the Internet also exists in the IoT because the Internet is the core and center of the IoT. Compared to other traditional networks, IoT nodes have low capacity and limited resources and lack manual control. The rapid growth and widespread adoption of IoT devices in daily life also make IoT security issues a big concern, which raises the need to develop network-based security solutions. While current systems work well in identifying some attacks, others are still difficult to detect. With the growth of network attacks and the massive increase in the amount of information present in networks, faster and more efficient attack detection methods are required [2] and there is no doubt that there is room for more progressive methods to improve network security. . In this context, in order to provide embedded intelligence in the IoT environment, we can consider Machine Learning (ML) as one of the most effective computing models. Machine learning approaches have been used for various network security tasks such as network traffic analysis [3],[4],[5], intrusion detection [6], and botnet detection [7]. Machine learning can be applied to the task of attack detection through two main types of cyber analysis: signature-based (sometimes also called exploit-based) or anomaly-based. Signature-based techniques are designed to detect known attacks using specific characteristics of the traffic (also known as "signatures") in those attacks. One of the advantages of this class of detection technique is its ability to efficiently detect all known attacks without generating an overwhelming number of false positives. In the literature, some works use techniques based on signatures to detect attacks [3], [7]; for example, in the field of network traffic analysis [3] applied four different machine learning techniques as preliminary tools to learn the characteristics of some known attacks.

Signature-based techniques were also used in [7] to identify compromised machines by identifying botnet network traffic patterns. The main disadvantage of signature-based approaches is that effective use of these approaches requires frequent manual updates of attack signatures and that these approaches cannot detect previously unknown attacks. A second class of detection methods is anomaly-based detection. This class models normal network behavior and anything abnormal is considered an attack.

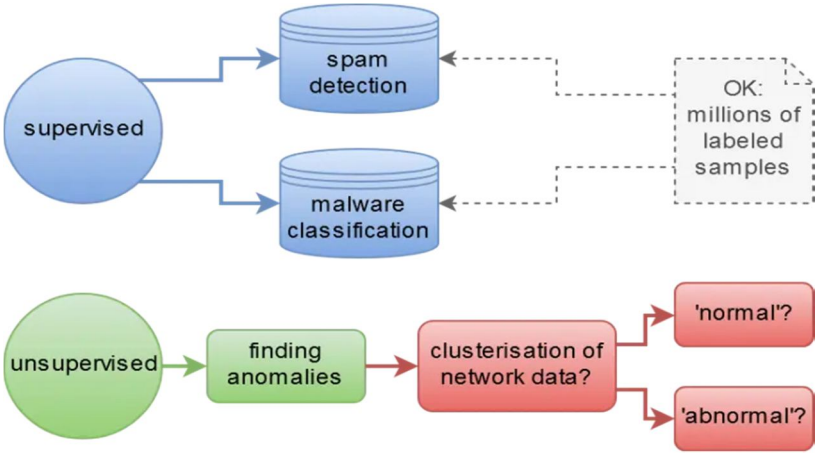



Fig I. Machine Learning and Cyberattacks

In this study, we contribute to the literature as part of the defense against IoT attack behavior by investigating the effectiveness of using machine learning approaches to detect IoT network attacks. The detection algorithms are evaluated using the latest Bot-IoT dataset, which combines legitimate and simulated IoT network traffic along with different types of attacks [9]. Using the Random Forest Regressor algorithm, features were selected from this dataset. In the implementation phase, seven different machine learning algorithms were used and high performance was achieved. The following are the machine learning algorithms we used: K-Nearest Neighbors (KNN), ID3 (Iterative Dichotomizer 3), Quadratic Discriminant Analysis (QDA), Random Forest, AdaBoost, Multilayer Perceptron (MLP), and Naıve Bayes (NB).

## II. LITERATURE REVIEW

The field of using machine learning has been extensively researched in the past [6] and several scholarly papers have been published on intrusion detection using data mining techniques and machine intelligence [10]. However, most of these previous studies only used machine learning techniques for intrusion detection in traditional networks. In this study, we therefore extend this area of research with the specific application of machine learning to attack detection in the IoT context. The application of machine learning techniques in IoT is still in the early stages of research, specifically in the field of IoT security, but it has a huge potential to discover insights from IoT data [11]. In IoT networks, machine learning principles such as pattern recognition, anomaly detection, and behavior analysis can be used to detect potential attacks and stop abnormal behavior.

To review the recent research on machine learning attack detection in IoT networks, we have reviewed various studies and summarized them in Table I. Machine learning algorithms, datasets, and detection approaches are presented in each study. In selecting these studies, we focused on the use of different machine learning algorithms and datasets. Studies provide evidence that machine learning techniques can achieve success in detecting attacks. From the papers discussing the issue of using machine learning for IoT security, detection methodologies can be categorized as unsupervised methods [10], [12], [13], [14] and supervised methods [15], [16], [17], [9], [18].

Many studies have suggested that machine learning techniques can be used to support attack detection tasks, including kmeans, artificial neural networks (ANN), Random Forest (RF), auto-encoder, and others, and several authors have applied unsupervised machine learning algorithms to problems with detection. Autoencoders are some of the most prominent unsupervised algorithms that have been used in many works; for example Mirsky et al. [10] proposed the use of auto-encoders to extract features from datasets to improve cyber threat detection. They introduced Kitsune, an unsupervised network intrusion detection system that has the ability to learn to effectively detect network attacks. Kitsune's core algorithm (KitNET) uses a set of neural networks, known as autoencoders, to distinguish between normal and anomalous traffic patterns. In [12], Meidan et al. proposed and evaluated a new detection method that extracts behavioral snapshots from the network and also uses auto-encoders to detect abnormal network traffic from compromised devices.

The main disadvantage of using unsupervised machine learning algorithms for detection problems is that in network traffic, most flows are normal and anomalies such as attacks and outliers are rare, which negatively affects the success rate and anomaly detection. For this reason, better results are expected for supervised techniques. On the other hand, many supervised learning algorithms are used to detect attacks and are trained on datasets with labels indicating whether instances have been pre-classified as attacks or not. In [19], Elike Hodo used ANN and Support Vector Machine algorithms to detect attacks on non-Tor traffic using ML techniques on UNBCIC datasets. In order to accurately identify whitelisted IoT device types, in [15], a Random Forest algorithm was applied to features extracted from network traffic data. Recent work that takes a similar approach to our study was presented by Moustafa et al. [9] in the original paper that proposed the Bot-IoT dataset. They used LSTM, SVM, and RNN machine learning models to evaluate the IoT dataset, but did not determine the outrageous robustness of their models in their analysis. In our work, we use the same Bot-IoT dataset presented in [9], but we focus on extracting new features from the dataset and evaluating different machine learning algorithms on this dataset. [22] is another study that used the BoT-IoT dataset.

## III. PROPOSED MODEL

We chose the Bot-IoT dataset for experiments due to its regular updates, wide variety of attacks, inclusion of IoT-generated traffic, and ability to generate new features from the raw dataset. The Bot-IoT dataset [10] was created at the Cyber Range Lab at the Australian Center for Cyber Security (ACCS). This dataset has three main types of attacks that are based on botnet scenarios such as Probing, DoS and Information Theft. We used CICFlowMeter to extract flow-based features from raw traffic paths. CICFlowMeter [26] is a network traffic flow generator distributed by CIC for generating 84 network traffic functions.

### *A. Dataset*

As applications for various network security tasks use machine learning methods, large data sets are needed to analyze network flows and distinguish between normal and abnormal traffic. Several experiments have been done over the years to create network datasets. As shown in Table I , most studies using machine learning have tested their work on simulated or real network data. Although much of these datasets remain private, mainly due to security concerns, some have become publicly available, such as DARPA 98, KDD99, UNSW-NB15, ISCX, CICIDS2017, and N-BaIoT. Although several datasets have been created, the development of realistic IoT and network traffic datasets that include new botnet scenarios is still scarce. More importantly, some datasets lack the inclusion of IoT-generated traffic, while others neglect to create any new features. In some cases, the test environment used was not realistic, while in other cases the attack scenarios were not sufficiently diverse. For example, in [12], Meidan et al. created a publicly available IoT dataset called N-BaIoT, and many later studies have used this dataset to train and test their classification models. Although this dataset is relatively large and clean, it is unbalanced and the ratio of normal data is much lower compared to attack data. Moustafa et al. [9] tried to eliminate the shortcomings by designing the Bot-IoT dataset that we used for our experiments. The Bot-IoT dataset includes legitimate and simulated IoT network traffic along with various types of attacks[14]. BotIoT database attacks are divided into three types: probing, DoS and information theft attacks

### *B. Machine Learning Algorithms*

We used the Bot-IoT dataset to evaluate seven well-known machine learning classifiers: (K-Nearest Neighbors (KNN), ID3 (Iterative Dichotomizer 3), Random Forest, AdaBoost,
Quadratic Discriminant Analysis (QDA), Multilayer Perceptron (MLP) and Naıve Bayes (NB). When choosing these classifiers, emphasis is placed on combining popular algorithms with different characteristics. In this context, the algorithms used are briefly examined below.

### *C. Implementation Process*

Our method consists of five basic steps: feature extraction, data preprocessing, data partitioning, feature selection, and implementation of machine learning algorithms.

1) *Feature Extraction:* CICFlowMeter [25] was used to extract flow-based features (in pcap format) from raw network traffic data. CICFlowMeter is a network traffic flow generator distributed by CIC that generates 84 network traffic characteristics. It reads the pcap file and creates a visual document of the extracted features and also offers a csv file of the dataset. This process was primarily designed to improve the predictive capabilities of the classifiers by extracting new features of the dataset.

2) *Data Preprocessing:* Transformation data preprocessing operations are used to transform a dataset into a structure suitable for machine learning. This step also involves cleaning the dataset by removing irrelevant ones or corrupted data that can affect the accuracy of the dataset, making it more efficient.
3) *Data Partitioning:* During the machine learning process, data is needed for learning to take place. In addition to the data needed for training, test data is needed to evaluate the algorithm's performance to see how well it performs. In our study, we considered 80% of the Bot-IoT dataset as training data and the remaining 20% as testing data.
4) *Feature Selection:* It is important to reduce the number of features and only use features needed to train and test algorithms to find a lightweight security solution suitable for IoT systems [13]. We used the Random Forest Regressor algorithm as a feature selection technique. The random forest regressor has proven to be an effective method of reducing the dimensions of the data set. By reducing the input data features from more than 80 network traffic features to 7, the model will train and respond faster. The feature importance weights for the entire data set are shown in Fig. 2.
5) *Implementation Of Machine Learning Algorithms*: All experiments were performed in Python using Python machine learning libraries (scikit-learn, Matplotlib, Pandas and NumPy). We organized the evaluation of machine learning algorithms for the dataset in three stages: applying the proposed algorithms to each attack in the dataset separately; applying algorithms to the entire data set with a set of features combining the best features for each attack (a list of these features is given in Table II); and applying the algorithms to the entire data set with the seven best features obtained in the feature selection step.

## IV. RESULTS

### A. Metrices

When evaluating the performance of machine learning models, it is crucial to define performance measures that are appropriate for the task at hand. To evaluate our results, we used the most important performance indicators for accuracy, precision, f-measure and recall as shown in the equations below:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

$$F - measure = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \quad (4)$$

### B. Results

As mentioned in the previous section, we organized the evaluation of the machine learning algorithms for the dataset in three stages as follows. Phase 1: applying machine learning algorithms to each attack in the dataset separately; Phase 2: applying machine learning algorithms to the entire dataset with a set combining the best features for each attack; and Phase 3: applying machine learning algorithms to the entire data set with the seven best features obtained in the feature selection step. The results of all experiments are presented in the following tables. The performance evaluation procedures were repeated 10 times for each machine learning algorithm, and the numbers shown in the tables represent the arithmetic averages of these 10 processes.

Phase 1: apply machine learning algorithms to each attack in the dataset separately. Seven different machine learning methods are applied to 10 different types of attacks, and the results are shown in Table I.

Table I. Distribution of Results According To Type of Attack

| Attack Names | F-Measures | | | | | | |
|---|---|---|---|---|---|---|---|
| | NB | QDA | RF | ID3 | AB | MLP | KNN |
| DDOS HTTP | 0.72 | 0.85 | 0.96 | 0.96 | 0.96 | 0.95 | 0.96 |
| DDOS UDP | 0.73 | 0.92 | 0.98 | 0.98 | 0.98 | 0.97 | 0.98 |
| DDOS | 0.71 | 0.85 | 0.99 | 0.99 | 1.0 | 0.74 | 0.99 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| TCP | | | | | | | |
| DOS HTTP | 0.72 | 0.82 | 0.95 | 0.96 | 0.95 | 0.95 | 0.96 |
| DOS UDP | 0.72 | 0.83 | 0.97 | 0.97 | 0.98 | 0.98 | 0.97 |
| DOS TCP | 0.64 | 0.74 | 1.0 | 1.0 | 1.0 | 0.78 | 0.99 |
| Data exfiltration | 0.72 | 0.76 | 0.96 | 0.97 | 0.97 | 0.94 | 0.97 |
| Keylogging | 0.72 | 0.82 | 0.95 | 0.95 | 0.95 | 0.91 | 0.98 |
| Service Scan | 0.73 | 0.83 | 0.95 | 0.95 | 0.95 | 0.94 | 0.94 |
| OS Scan | 0.72 | 0.76 | 0.94 | 0.97 | 0.98 | 0.97 | 0.99 |

If there is equality in the F measure in the results of the algorithms, the following values are examined to rule out equality: accuracy, precision, recall, and time.

Observing the results , it can be noted that all algorithms, except Naive Bayes (NB) and Quadratic algorithm (QDA), achieved more than 90% success in detecting almost all types of attacks. The ID3 algorithm was the most successful algorithm, completing 6 of the 10 tasks (DDOSHTTP, DDOS-UDP, DOS-HTTP, DOS-TCP, Data Exfiltration, and Service Inspection) with the highest scores. In fact, for everyone tasks, ID3 shares its highest score with at least one other algorithm. However, its low processing time puts it ahead of other algorithms. The last algorithm used in all problems was Naive Bayes, the algorithm with the lowest F-measure. Especially with

DOS TCP attack, had a relatively low score. Although Naive Bayes performed worse than the other algorithms, it was much better than the alternatives in terms of speed. However, it is also necessary to mention QDA here because QDA had the second worst performance among the algorithms.

Phase 2: Applying machine learning algorithms to the entire dataset with a set of features that combine the best features for each attack. The entire data set is used in this phase. Seven different machine learning methods were implemented on the entire dataset and we used feature sets that were extracted for each attack separately. Table II shows the results obtained using the 13 features extracted for the attacks.

Table II. Implementation of Features Obtained From Phase1

| ML Algorithm | Accuracy | Precision | Recall | F-Measure | Time |
|---|---|---|---|---|---|
| NB | 0.78 | 0.84 | 0.78 | 0.75 | 5.056 |
| QDA | 0.88 | 0.89 | 0.88 | 0.87 | 6.1964 |
| RF | 0.98 | 0.98 | 0.98 | 0.98 | 27.0328 |
| ID3 | 0.99 | 0.99 | 0.99 | 0.99 | 19.3447 |
| Adaboost | 1.0 | 1.0 | 1.0 | 1.0 | 308.9403 |
| MLP | 0.84 | 0.88 | 0.84 | 0.83 | 1011.5001 |
| KNN | 0.99 | 0.99 | 0.99 | 0.99 | 2052.1801 |

Observing Table III, it can be seen that the best performing algorithm was Adaboost, followed by KNN and ID3. ID3 is noticeably faster than KNN, so it takes precedence for this feature. The lowest scoring algorithm was Naive Bayes with a score of 0.75. In terms of speed, NB and QDA were the fastest. Although KNN had a high performance score, it was still noticeably slower than the other algorithms.

Stage 3: applying machine learning algorithms to the entire dataset with the seven best features obtained in the feature selection step. In terms of F-measure, there was no significant change in the performance of the algorithms, but in terms of speed, the running times of all algorithms were noticeably reduced. The reason for this reduction in execution time is that 13 attributes are used in the method used in Table IV, while only 7 attributes are used in Table III. This reduction in the number of features reduced the running time of the machine learning algorithms.

Table III. Implementation of Features Obtained Using Random Forest Regressor For All Dataset

| MLAlgorithm | Accuracy | Precision | Recall | F-Measure | Time |
|---|---|---|---|---|---|
| NB | 0.79 | 0.85 | 0.79 | 0.77 | 4.0472 |
| QDA | 0.87 | 0.89 | 0.87 | 0.86 | 4.4056 |
| RF | 0.97 | 0.97 | 0.97 | 0.97 | 28.9246 |
| ID3 | 0.97 | 0.97 | 0.97 | 0.97 | 17.0899 |
| Adaboost | 0.97 | 0.97 | 0.97 | 0.97 | 238.8618 |
| MLP | 0.84 | 0.87 | 0.84 | 0.83 | 949.6977 |
| KNN | 0.99 | 0.99 | 0.99 | 0.99 | 1615.9852 |

The final results of the implementation (see Table VI) are compared with the study in the literature. For this comparison, a study by Ferrage et al. [14] in 2019 he was selected. This is because the mentioned work used the same data set and also two machine learning methods similar to the ones we used. These similar machine learning algorithms are Random Forest and Naive Bayse. The key difference between our work and theirs is the feature set used. They used the original feature set while we used the new feature set extracted by CICFLOWMETER. The detection rate (Recall) was determined as the main evaluation criterion. Table VI shows a comparison of the results obtained from the two studies. When the results are examined, it can be seen that the Random Forest algorithm used in our study is superior to the algorithm used in [14], and the same can be seen for most types of attacks with the NB algorithm. So we can see that the new features used in our work have increased the performance of both algorithms

Table IV. Comparison of Performance of The Two Algorithms

| Attack Names | Ferrag et al[14] | | Our Work | |
|---|---|---|---|---|
| | RF | NB | RF | NB |
| DDOS HTTP | 82.26% | 50.78% | 96% | 71% |
| DDOS TCP | 88.28% | 78.67% | 99% | 70% |
| DDOS UDP | 55.26% | 78.50% | 98% | 72% |
| DOS HTTP | 82.20% | 68.68% | 95% | 71% |
| DOS TCP | 81.77% | 65.56% | 100% | 63% |
| DOS UDP | 82.99% | 100% | 97% | 71% |
| Data exfiltration | 86.55% | 66.55% | 96% | 71% |
| Keylogging | 70.12% | 65.62% | 95% | 71% |
| OS Scan | 82.20% | 68.68% | 94% | 70% |
| Service Scan | 69.82% | 65.21% | 95% | 72% |

## V. CONCLUSIONS

This paper focused on the detection of IoT network attacks using machine learning methods. In this context, Bot IoT [9] was used as a dataset due to its regular updates, wide variety of attacks, and different network protocols. We used CICFlowMeter[25] to extract flow-based features from raw traffic routes. CICFlowMeter generates 84 network traffic feature datasets that define network flow. During implementation, importance weight calculations were performed using the Random Forest Regressor algorithm to decide which features would be used in machine learning methods. Two approaches were used in performing these calculations.

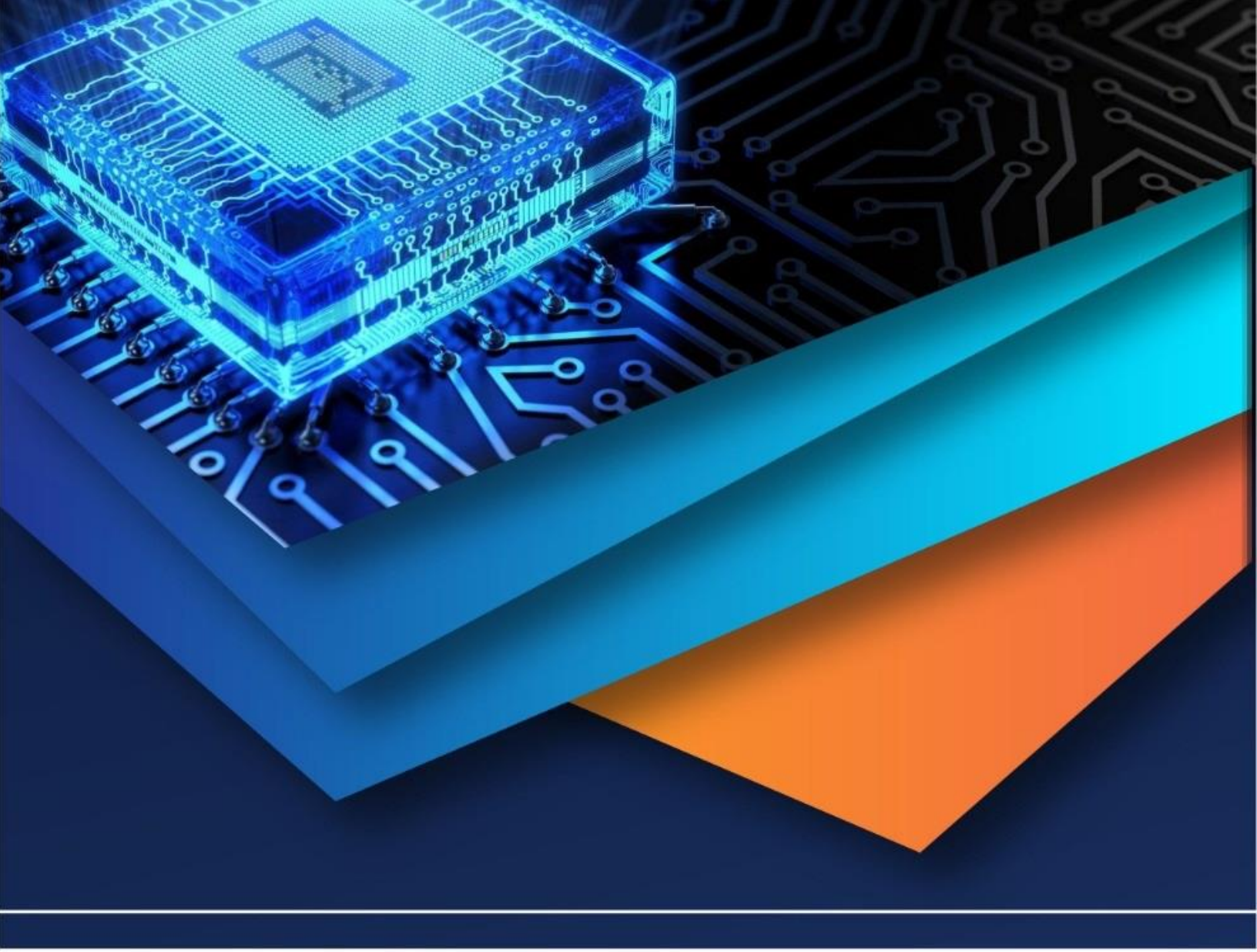
In the first approach, importance weights were calculated separately for each type of attack, and in the second approach, all attacks were collected in one group and importance weights were calculated for that group; i.e., common characteristics that were important to all attacks were determined.

Finally, seven machine learning algorithms that are widely used and of varying quality were applied to the data. These algorithms and the achieved performance ratios according to F-measure are as follows: F-measure had a value between 0 and 1; Naive Bayes was 0.77; QDA was 0.86; Random Forest was 0.97; ID3 was 0.97; AdaBoost was 0.97; MLP was 0.83; and K Nearest Neighbors was 0.99.

In this research, we investigated seven supervised algorithms. As a future work, it would be interesting to evaluate the performance of some unsupervised algorithms. In addition, we applied different machine learning algorithms independently. In the future, we would like to combine different machine learning algorithms as a multi-layer model to improve the detection performance.

## REFERENCES

[1] J. Deogirikar and A. Vidhate, "Security attacks in iot: A survey," International Conference on I-SMAC (I-SMAC), pp. 32–37, 2017.
[2] T. Bodstrom and T. H¨ am¨ al¨ ainen, "State of the art literature review¨ on network anomaly detection with deep learning," Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pp. 64–76, 2018.
[3] Arnaldo, A. Cuesta-Infante, A. Arun, M. Lam, C. Bassias, and K. Veeramachaneni, "Learning representations for log data in cybersecurity," International Conference on Cyber Security Cryptography and Machine Learning, pp. 250–268, 2017.
[4] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285–1298, 2017.
[5] J. Radford, B. D. Richardson, and S. E. Davis, "Sequence aggregation rules for anomaly detection in computer network traffic," arXiv preprint arXiv:1805.03735, 2018.
[6] Lambert and M. Glenn, "Security analytics: Using deep learning to detect cyber attacks," 2017.
[7] M. Stevanovic and J. M. Pedersen, "Detecting bots using multi-level traffic analysis." IJCSA, vol. 1, no. 1, pp. 182–209, 2016.
[8] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource iot devices: A game-theoretic methodology," IEEE International Conference on Communications (ICC), pp. 1–6, 2016.
[9] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, 2019.
[10] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," arXiv preprint arXiv:1802.09089, 2018.
[11] X. Yuan, C. Li, and X. Li, "Deepdefense: identifying ddos attack via deep learning," IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–8, 2017.
[12] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018.
[13] M. K. Putchala, "Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru)," 2017.
[14] M. A. Ferrag and L. Maglaras, "Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," IEEE Transactions on Engineering Management, 2019.
[15] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized iot devices using machine learning techniques," arXiv preprint arXiv:1709.04647, 2017.
[16] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques," International Conference on Mobile Networks and Management, pp. 30–44, 2017.
[17] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Rule generation for signature based detection systems of cyber attacks in iot environments," Bulletin of Networking, Computing, Systems, and Software, vol. 8, no. 2, pp. 93–97, 2019.
[18] V. H. Bezerra, V. G. T. da Costa, S. B. Junior, R. S. Miani, and B. B. Zarpelao, "One-class classification to detect botnets in iot devices," Anais do XVIII Simposio Brasileiro em Seguranc¸a da Informac¸´ ao e˜ de Sistemas Computacionais, pp. 43–56, 2018.
[19] Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–6, 2016.
[20] H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for internet of things devices," IEEE 34th international performance computing and communications conference (IPCCC), pp. 1–8, 2015.
[21] Y. Yavuz, "Deep learning in cyber security for internet of things," Ph.D. dissertation, 2018.
[22] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in iot networks," arXiv preprint arXiv:1905.05137, 2019.
[23] Cvitic, D. Perakovi´ c, M. Peri´ sa, and M. Botica, "Novel approach for˘ detection of iot generated ddos traffic," Wireless Networks, pp. 1–14, 2019.
[24] Z. A. Baig, S. Sanguanpong, S. N. Firdous, T. G. Nguyen, C. So-In et al., "Averaged dependence estimators for dos attack detection in iot networks," Future Generation Computer Systems, vol. 102, pp. 198– 209, 2019.
[25] H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features." ICISSP, pp. 253–262, 2017.
[26] S. Yu, "Study on the internet of things from applications to security issues," International Conference on Cloud Computing and Security, pp. 80–89, 2018.
[27] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," Emerging artificial intelligence applications in computer engineering, vol. 160, pp. 3–24, 2007.
[28] K. Kostas, "Anomaly detection in networks using machine learning," Ph.D. dissertation, 08 2018.
[29] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," International journal of computer science and network security, vol. 7, no. 12, pp. 258–263, 2007.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY