JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
ICASET
2024

Research Article
Collection: Intelligent Computing of Applied Sciences and Emerging Trends

Lightweight Intrusion Detection for IoD Infrastructure using Deep Learning

Hafiz Muhammad Sanaullah Badar[1], Nadeem Iqbal Kajla[1*], Jehangir Arshad[2], Najia Saher[3], Manal Ahmad[1], and Muhammad Ahsan Jamil[1]

[1]Institute of Computing, Muhammad Nawaz Sharif University of Agriculture, Multan, 60000, Pakistan.
[2]Department of Electrical and Computer Engineering, COMSATS University Islamabad, Pakistan.
[3]The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan.
[*]Corresponding Author: Author's Nadeem Iqbal Kajla. Email: nadeem.iqbal@mnsuam.edu.pk

Academic Editor: Salman Qadri **Published:** April 01, 2024

Abstract: The rapid growth of the Internet of Drones (IoD) has created new challenges for cybersecurity experts. Network intru- sion remains a major concern in cyberspace, and traditional in- trusion detection methods are limited in their ability to detect and prevent attacks. Machine learning-based approaches have shown promise in detecting network intrusions, but their accuracy is still a challenge. To address this, a machine learning approach was proposed using seven classifiers, including DT, random forest, na¨ıve bayes, Adaptive Boosting Algorithm (ADA), Adaptive Boosting Algorithm (XGB), K-Nearest Neighbors (KNN), and logistic regression. The proposed model was evaluated on the CICIDS2017 dataset, achieving high accuracies with the DT classifier having the highest accuracy of 0.99. This approach can be applied to detect and prevent network intrusions in the growing IoD network, ensuring the integrity, confidentiality, and availability of communication networks.

Keywords: Machine Learning; Intrusion Detection; Classification; naive Bayes; Deep Learning; cyber security; IOT; IoD.

1. Introduction

The evolution of the Internet of things (IOT) has run to the rise of a new application realm: the Internet of Drones (IoD) IOD is a rapidly mounting arena that syndicates unmannered vehicles (UAVs) with devices to deliver innovative solutions for different industries, some of them are transportation, agriculture, and surveillance. The combination of IOT and Drones enable a new variety of potential with real-time monitoring, self-directed operations, and remote sensing. However, with the rapid expansion of IoD , there are some prominent security challenges that need to be talked. IOT leading to a captivating grouping with drones known as the Internet of drones (IoD). This fusion offerings thrilling possibilities, but it also have some considerable challenges that require to be devotion [1].

The Major concern among all is the security of data. This include protecting against the data breaches, cyber-attacks and privacy abuses. There are several major risk in the field of IoT And IoD, for example just imagine someone breakdown your computer and thieving your personal information.

Denial of services (DoS) attack, is another major challenge which re like digital traffic jams that can involve to interrupt IOT and IoD Devices. Encryption is used to protect data against these attacks in data transmission.[2-5]. However encryption can also produce problems by triggering conflicts during critical task, when there are several requests for secure transmission. In the IoT world , there a risk of intruders exploiting system weakness, which is like someone looking an open door of your home. This can lead to inefficiencies and interrupt resource operations. For example, interfering with delivery system of drones can cause it to drops its cargo in the wrong destination [6-7].

To overcome these challenges, an intrusion Detection System(IDS) is hired, that act as a digital security guard , that monitor the network for unauthorized and harmful traffics and activities. Its like having a security camera at your home door to detect any suspected behavior. In Kernel, the mixing of IoT

and drones into IoD proposed exciting prospects but also concerned security challenges , to protect your home and personal information online. To ensure the accurate working and safety of IoT and IoD, we regularly take security measures, for example updating home security system, to stay safe. The key aspect of an Intrusion detection system (IDS), is to mark and identify the unauthorized access attempts as much as possible. Hijacking a drone is also a major security concern that can be resolved by detecting unauthorized attention and taking necessary measure on right time through intrusion detection system [8-11]. In this aspect intrusion detection system (IDS) become necessary to secure the IoD Networks. IDS can detect and take necessary measures to prevent unlicensed access to network element and ensure the integrity, confidentiality and availability of network [12]. The size and complexity of IoD grows, so traditional IDS are insufficient. Cyber security policies are developing more intelligent approaches by using machine learning algorithms because blacklisting now not enough in preventing the phishing attacks. BY identifying the cyber-attack and warning the security system prevention response, machine learning combat this problem [13].

The proposed work contain the objective, to develop an automated signature generation method for network intrusion detection (NID) in IoD using machine learning algorithms. The IDS should have the features of effectiveness, adaptability and extensibility [14]. The regular updates to its database of rules are required for a signature based IDS, that is time consuming and costly. In the proposed method machine learning algorithms are used to generate an optimized set of intrusion detection rules using features selection techniques to reduce the space of search and help to find suitable attributes [15]. IoD networks are used in wide range of application including agriculture, transportation and surveillance, so the the security of IoD network is most important and critical. All risks associated with cyber-attack can cause harmful results including data breaching, loss of personal information and maybe physical harm. The proposed method automate the procedure of signature generation, keep the signature database current and reduce the harm to authentic users [16]. Thus the grouping of IoT and Drones has developed a new and wide range of possibilities, that helping real time monitoring, automate the operations, and remote sensing, but there are a lot of security risks that need to be addressed. Intrusion detection system with machine learning approach has become essential for the security of IoD networks. The proposed work, using machine learning approaches to automate signature generation for NID networks in IoD, can provide and optimized, effective and adoptable solution for the security of IoD networks.

1.1. Motivations and Contributions

With the increasing sophistication of cyberattacks, traditional methods of intrusion detection are proving to be insufficient in securing computer networks. Machine learning-based approaches have emerged as a promising solution, but their accuracy remains a challenge. We proposed machine learning approaches that uses seven classifiers to detect and classify network intrusions. The proposed approach is evaluated on a comprehensive dataset, and the results demonstrate its effectiveness in detecting network intrusions. The findings of this study have practical implications for cybersecurity experts and suggest that a machine learning approaches can enhance the accuracy of intrusion detection systems. This paper's key contributions can be summarized as follows:

1) Introducing machine learning approaches for intrusion detection and classification using machine learning models.
2) Employing seven different machine learning classifiers (DT, Random Forest, Naive Bayes, ADA, XGB, KNN, and Lgr) for detecting and classifying intrusions.
3) Evaluating the accuracy of the models and reporting other statistical metrics such as precision, recall, and F1-score.

The organization of this paper is as follows: In section III a detail review of related work and literatures given. In section IV describe the design of proposed work. In Section V, convey implementation details, results, and discussion. In the last Section V-E contains our conclusions.

2. Related Work

In most recent years, we have observe an alarming rise in cyber-attack that are highly damaging. This provoke conventional intrusion detection system (IDSs) should enhance their features and ability to detect and prevent the cyber-attack by using machine learning and deep learning. A lot of researchers tried with

their own ML/DL –based IDS solution using different technique and datasets for valuation and improvement.

For example, Shfaq et al. [17] presented a Semi-Supervised Learning (SSL) approach that utilizes a single hidden layer feedforward neural network (SLFN) and a sample categoriza- tion method to identify network anomalies. Singh et al. [18] proposed a peer-to-peer (p2p) anomaly detection system that is scalable and utilizes the RF (RF) technique to identify network anomalies. Tuan et al. [19] proposed an unsupervised learning method that utilizes Local Outlier Factor (LoF) to detect network attacks such as DDoS attacks in SDN. Ali et al. [20] introduced a three-tier intrusion detection and prevention system (IDPS) to detect Distributed Denial of Service (DDoS) attacks in SDN. Moustafa et al. [21] proposed an architecture that uses the Outlier Gaussian Mixture (OGM) scheme to detect web attacks.

After the study of existing contribution [17],[22],[23], we noticed that some solutions [21],[24], are not able to provide sufficient accuracy in IDS, that can cause serious consequence in a cyber-security system. It is necessary that IDS should provide high accuracy in detecting and prevention cyber-attack, because inaccurate IDS results leads you to false positive or false negative. The false positive may create unnecessary overheads that generate false alarm and it effect the confidence of cyber security system. On contrary false negative may lead you to security breach that goes undetected and keep the security system for more attacks.

In order to elaborate the low accuracy issue in intrusion detection system, we proposed an IDS that would prove high level of accuracy to detect and prevent cyber attack. The table 1 shows a comparative analysis of Machine learning and deep learning base intrusion detection system research of different authors that highlight the method, and associated benefits and drawbacks of intrusion detection system. The authors used a wide range of methods from semi supervised learning (SSL) to peer-to-peer approaches to elaborate the network security challenges.

3. Proposed Framework

A drone base network is used to design the proposed framework for Intrusion Detection (IoD). The system design I which several drones are interconnected with a central base station through internet. These drones collect the data from environment and send it to base station for further processing.

For Intrusion detection, first the collected will be classified using different machine learning approaches. There are seven classifiers DT, RF, Naïve Bayes, ADA, XGB, KNN, and logistic regression are used to classify the data. After classification, data is transferred to data center for storage and analysis.

The system model is design in a way that it can provide active and streamlined method for identifying and categorizing network intrusion in drone based network. As stated above accuracy is crucial for maintaining the security and integrity of network, the proposed model is capable enough that system can provide high accuracy by utilizing this proposed classifiers.

3.1. System Design

A comprehensive and widely known cyber security dataset, CICIDS2017 is used in this project. Our system design key components are starting with python environment setup using the libraries such as Numpy, Pandas, Scikit-learn, and tensor Flow (or PyTorch) for machine learning. The library Pandas is used to load CICIDS2017 data set and Processed through the data processing. The data cleaning, tackle the missing Values, outlier handling and scaling the features is done for the uniformity of data. The dataset Split into two subsets one is for training and other is for testing.

The 7o percent dataset is used for training and 30 percent is used for testing. For the optimization of our chosen model we conduct a hyper parameter search, which deal with algorithms like Logistic Regression, Decision Tree, Random Forest, Support vector machine, or deep learning algorithms and techniques. Model evolution done using the matrices like accuracy, precision, recall, F1-Score and ROC-AUC. At the end, using CICIDS2017 dataset, we deployed the optimized version of real time application that focus on the robust analysis of network intrusion detection.

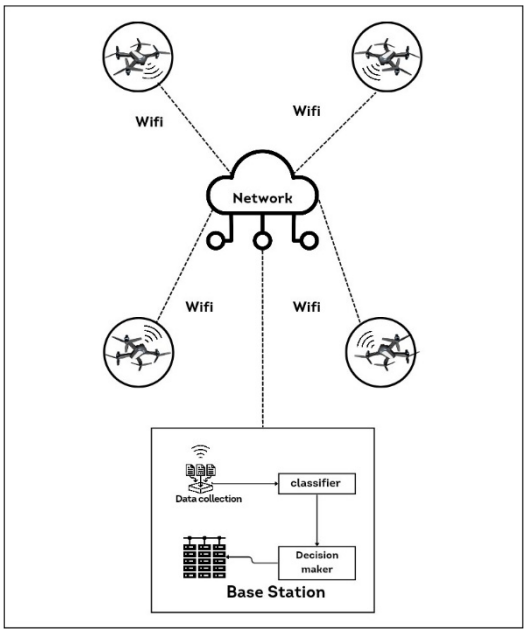

Figure 1. IoD Infrastructure

3.2. Dataset

The CICIDS2017 dataset gain high attention the research community since it was initiated, that make it necessary for the development of models and algorithms related to network security and intrusion detection [18]. This dataset contain the collection of normal and attack traffic of network traffic data, spanning a period of five days. The Canadian Institute of Cybersecurity has generate and accurately curated for research community.

As detailed by the dataset's author [26], The data CICIDS2017 is consist of 3,119,345 instances, each instance is presented by 83 attributes and related with one of 15 class labels. There are instances with missing class labels with incomplete details, total 288,602 instances have missing class labels and 203 instances have incomplete details [27]. A severe data preprocessing step was taken, to ensure the dataset's integrity and usability, which help to remove instances with missing or incomplete details.

Table 1. Comparative Analysis of ML/DL-Based IDSS

Sr#	Author	Techniques	Benefit	Drawback
1	Shfaq et al. [10]	Semi-Supervised Learning (SSL)	Improved anomaly detection	Limited to SSL approach
2	Singh et al. [11]	Peer-to-peer (p2p)	Scalability	Limited to RF technique
3	Tuan et al. [12]	Unsupervised Learning	DDoS attack detection	Limited to LoF
4	Ali et al. [13]	Three-Tier IDPS	DDoS attack detection	SDN-specific
5	Moustafa et al. [14]	Outlier Gaussian Mixture (OGM)	Web attack detection	Limited to web attacks

Succeeding this preprocessing, CICIDS2017 was converted to a consolidated version that contain 2,830,540 instances, ready for model development and analysis. An examination for redundant case was conducted in data quality processing but no duplicate instance was found. A foundation for research and experimentation was established on the resulting dataset for intrusion detection and network security. Table 2 and Table 3 conclude the properties of combine dataset and occurrence of every class label.

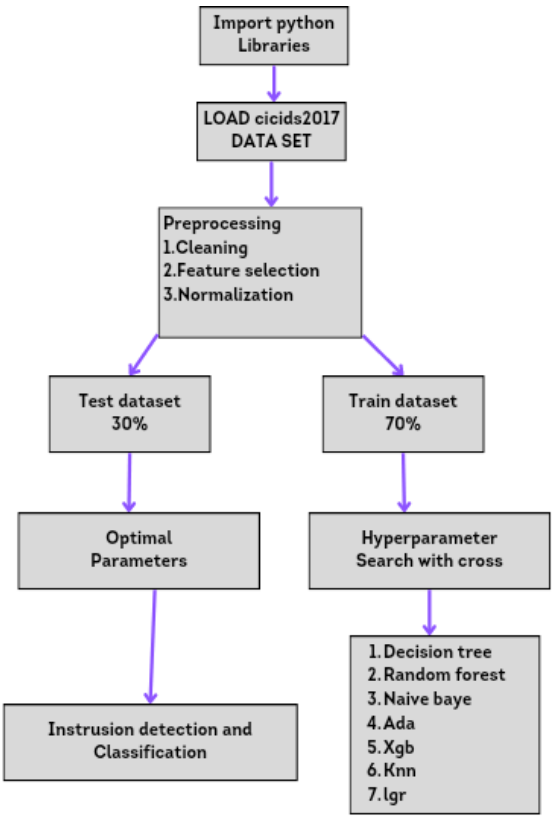

Figure 2. Flow diagram

Table 2. Properties of CICIDS2017 DATASET

Property Name	CICIDS2017
Type	Multi-class
Releasing year	2017
No.of distinct instances	2830540
Total features	83
No.of Classes	15

Table 3. Class-wise instance occurrence of cicids2017 DATASET

Sr.	**Class Name**	**Total No.of Instances**
1	Benign	, 891074
2	Portscan	158930
3	DDoS	128027
4	Bot	1966
5	Web attack-Brute force	R1507
6	Web attack- XSS	652
7	Infiltration	36
8	Web attack-sql injection	21

3.3. Dataset Pre-Processing

In the domain of machine learning, data preprocessing is a difficult step that includes preparing and transforming raw data into a format that can be utilize for advance analysis. In article, the dataset experienced several preprocessing steps to make it for training and testing. Initially, normalization was utilize to standardize the data and values. Datasets cleaning and feature selection were also accomplished to conform that the data was correct and relate to the problem at hand. Later on, the preprocessed datasets

was split into training and testing datasets, with a split of 70% and 30% respectively. The datasets contain 8 classes, and the no of instances compared to every class is shown in table 3.

Characteristics extraction is an important preprocessing task that includes selecting related structures from the data to assemble the training and testing datasets for the algorithm. This steps perform a critical part in summarize the possibility overfitting, constructing analysis more straightforward, and upgrade the generalization of the model. It is significant to take note on the improper feature extraction can create the model to take extended to train and test, as it requires to undergo through further data than necessary.

Subsequent to feature extraction, the training datasets was fed into the machine learning method to form a model. Based on the algorithm, this step may be created once or numerous times to ameliorate the model precision. Lastly, the test dataset was utilized to estimate the model's accuracy. The model performs better when it is closer to 100% in accuracy. Although, it is important to pay attention on that it is crucial to achieve 100% accuracy because of the increasing scope of data and the model go through from variations.

3.4. System Specifications

Python, along with its extensive libraries, played a pivotal role in the practical implementation of our proposed model. The system used for training and testing the model was equipped with robust hardware specifications, featuring four central processing units (CPUs) with a clock speed of 2.5 GHz and a generous 32GB of random access memory (RAM). The software stack included Python 2.7, a well-established version, and the Windows operating system.

The complete workflow of the tool employed for implement- ing and evaluating our model is visually illustrated in Figure 3. This workflow diagram outlines the step-by-step process, from data preprocessing to model training and evaluation, providing a clear overview of how the model was developed and assessed.

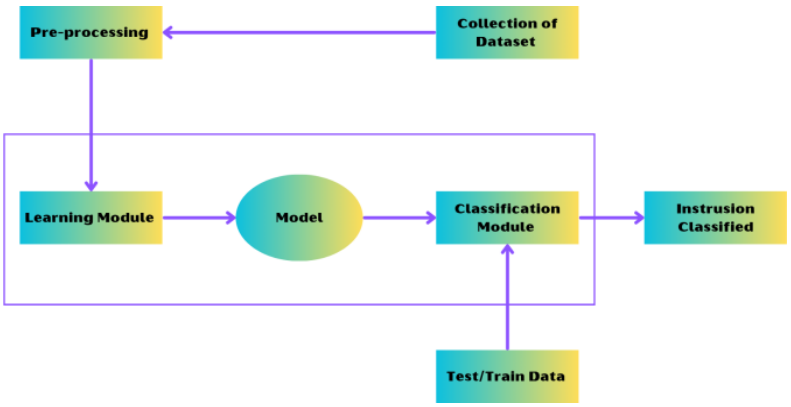

Figure 3. Workflow

4. Results and Discussion

In this section, we will provide a brief overview of the outcomes of the applied methodology. The testing and training process, as well as the implementation of the proposed approach, will be discussed. In order to classify network intrusions, a total of seven classifiers were utilized as part of the machine learning approach. We will now delve into the details of the results and methodology in the following sections. In 2012, a lightweight NIDS based on DTs (DT) [28] was proposed, in which information/ Gini entropy is used to make decisions based on a tree-like model. An event's potential consequences and outcomes can be generated using the model. The design requires little training time, but it suffers from overfitting. Multiple poor DT classifiers make up the RFs (RF) dependent NIDS [29]. In comparison to the DT design, the design can effectively handle the overfitting problem. For the classification function, Nave Bayes networks (NB) [24] were suggested. The NB model responds to the question "what is the likelihood of a specific form of attack being directed at the observed system?" [30]. Using a guided acyclic graph, the model defines the effect between neighboring nodes (DAG). Malicious executables were detected using the NB model [31]. For the evaluation, a total of 4,266 programs were used, with 3,265 malicious binaries and 1,001 clean binaries. According to their research, NB outperformed conventional rule- based/signature-based

structures. For NID, Adaptive Boosting (Adaboost) [32] was used. Adaboost, like RF, is built on a large number of weak classifiers.

4.1. Evaluation Matrix

The proposed model employs a variety of classifiers for Network Intrusion Detection (NID) and classification. These classifiers rely on several statistical equations to determine their performance and effectiveness in distinguishing between normal and intrusive network activities. Here's a more detailed explanation of the key statistical values and metrics utilized in the evaluation process:

- **True Positive (TP):** TP represents the number of correctly identified instances where the model correctly classified an intrusion as an intrusion.
- **True Negative (TN):** TN corresponds to the number of accurately recognized instances where the model correctly classified non-intrusions as non-intrusions.
- **False Positive (FP):** FP is the count of instances where the model mistakenly classified a non-intrusion as an intrusion.
- **False Negative (FN):** FN denotes the instances where the model incorrectly classified an intrusion as a non- intrusion.

These statistical values (TP, TN, FP, and FN) form the basis for evaluating the model's classification performance. They are used to calculate the following key metrics:

- **Recall (R):** Recall, also known as sensitivity or true positive rate, quantifies the model's ability to correctly identify intrusions. It is computed using the following equation:

$$R = \frac{TP}{TP + FN} \quad (1)$$

- **Accuracy Rate (Acc):** The accuracy rate measures the overall correctness of the model's predictions. It is deter- mined by the equation:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

- **Precision (Pre):** Precision evaluates the model's capability to classify instances as intrusions accurately. It is calculated as:

$$Pre = \frac{TP}{TP + FP} \quad (3)$$

In situations where there's a need to strike a balance between precision and recall, the F1-score is employed as an evaluation metric. The F1-score considers both precision and recall, offering a single value to assess the model's overall performance. It is calculated as follows:

$$F1 = \frac{2 \cdot (Pre \cdot R)}{Pre + R} \quad (4)$$

These statistical values and metrics provide a comprehensive assessment of the model's ability to detect network intrusions while considering trade-offs between accuracy, precision, and recall.

4.2. Model evaluation on Training dataset

The performance of different machine learning algorithms for NID using the dataset CICIDS2017 shown in table 4. Precision, recall F1-Sccore, Accuracy and cross validation mean for evaluation. The findings indicate that both DT and ADA classifiers demonstrated perfect precision, recall, F1-score, and accuracy. Conversely, the naive Bayes algorithm exhibited the least favorable performance among all classifiers. Additionally, RF, XGB, and KNN classifiers delivered commendable results, boasting high precision, recall, F1-score, and accuracy. Notably, the KNN algorithm emerged with the highest mean during cross-validation.

Please be aware that the outcomes presented here stem from the analysis of training and validation datasets, and may not translate effectively to novel, unseen data. Hence, it is imperative to conduct

additional assessment and experimentation using independent datasets to confirm the efficacy of these algorithms. Nevertheless, the findings indicate that machine learning algorithms exhibit proficiency in identifying and categorizing network intrusions, with DT and ADA classifiers showing notable potential in this regard.

Table 4. Performance Evaluation of Different Classifiers on Training

No.	Algorithm	Precision	Recall	F1-Score	Accuracy
1	DT	1	1	1	1
2	Random Forest	0.99	0.99	0.99	0.99
3	Na¨ıve baye	0.86	0.84	0.82	0.85
4	ADA Classifier	1	1	1	1
5	XGB Classifier	0.97	0.97	0.97	0.97
6	KNN Classifier	0.98	0.92	0.98	0.98
7	Lgr Classifier	0.9	0.9	0.9	0.91

4.3. Model Evaluation on testing dataset

In Table 5 different classifier's performance evaluate on the testing dataset that shows important insights into the effectiveness to solve the given problem. The classifiers Decision Tree (DT), Random Forest (RF), Naïve Bayes, ADA Classifier, XGBoost(XGB) classifier, K-Nearest Neighbors (KNN), and logistic regression underwent a comprehensive assessment using crucial performance metrics such as Precision, Recall, F-1 Score and accuracy.

There are some notable trends in the results. The top performers are Decision Tree (DT), ADA Classifier, ad KNN. The top performer classifier achieved highest score across all key metrics, boasting perfect score of 1 for precision, recall, F1-score and accuracy for both DT and ADA. The KNN classifier also delivered 0.94 for precision, recall, F1-score and accuracy rating 0.98

On the other hand Random Forest (RF), XGBoost (XGB), and logistic regression classifier give a little lower performance then the leading classifier, their results was only acceptable. They continue to exhibit commendable performance on the testing dataset, making them viable choice depending on the specific problem requirements.

Table 5. Performance of different classifiers on testing

No.	Algorithm	Precision	Recall	F1-Score	Accuracy 1
1	DT	0.97	0.97	0.97	0.99
2	Random Forest	0.96	0.96	0.96	0.96
3	Na¨ıve Baye	0.86	0.84	0.81	0.85
4	ADA Classifier	0.96	0.96	0.96	0.97
5	XGB Classifier	0.96	0.96	0.96	0.96
6	KNN Classifier	0.96	0.96	0.96	0.98
7	Lgr Classifier	0.9	0.9	0.9	0.91

In contrast, the Naive Bayes classifier displayed the least favorable results, with lower scores across all metrics. This indicates that, in the context of the problem and dataset under consideration, the Naïve Bayes classifier may not be the most suitable choice.

In summary, the evaluation results highlight the Decision Trees (DT), ADA classifier, and KNN classifier as the most effective classifiers for this specific dataset. However, the ultimate choice of classifier should take into account the specific problem's nuances and the trade-offs between different evaluation metrics, as the optimal selection may vary depending on the context and objectives of the project.

4.4. Proposed Model Accuracy Comparison

The Figure 4 shows the accuracy of different classifiers for a given task. The classifiers include DT, RF, Naïve Bayes, ADA, XGB, KNN, and Logistic Regression. The accuracy of each classifier is presented in the second column of the Table VI. The DT classifier has the highest accuracy of 0.99, followed by KNN with 0.98 and ADA with 0.97. The Naïve Bayes classifier has the lowest accuracy of 0.85. Overall, the Table 6 provides a concise summary of the performance of different classifiers, which can be useful for selecting an appropriate classifier for a given task.

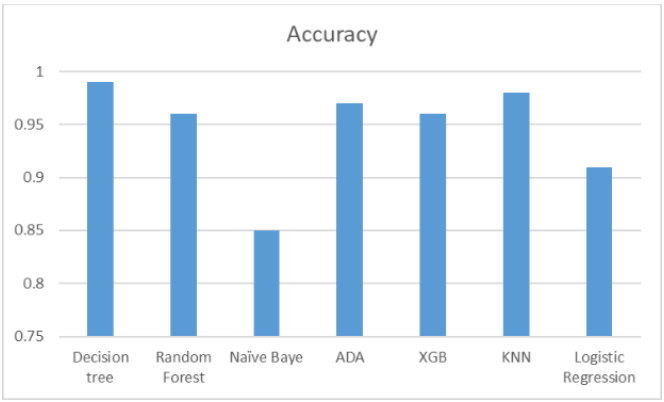

Figure 4. Accuracy Comparison

The Figure 4 shows the accuracy of different classifiers for a given task. The classifiers include DT, RF, Naïve Bayes, ADA, XGB, KNN, and Logistic Regression. The accuracy of each classifier is presented in the second column of the Table VI. The DT classifier has the highest accuracy of 0.99, followed by KNN with 0.98 and ADA with 0.97. The Naïve Bayes classifier has the lowest accuracy of 0.85. Overall, the Table VI provides a concise summary of the performance of different classifiers, which can be useful for selecting an appropriate classifier for a given task.

4.5. Comparative Analysis

As shown in Table 7, the proposed intrusion detection model using the Decision Tree (DT) algorithm on the CI- CIDS2017 dataset achieved an impressive accuracy rate of 0.99. This remarkable performance surpasses the majority of other models listed in Table 7, underscoring the efficacy of our proposed model in accurately detecting network intrusions.

Table 6. Classifier Performance

Classifier Name	Accuracy
DT	0.99
Random Forest	0.96
Na¨ıve Baye	0.85
ADA	0.97
XGB	0.96
KNN	0.98
Logistic Regression	0.91

Furthermore, the results also reveal that various machine learning models, including Random Forest (RF), Deep Belief Networks, Convolutional Neural Networks, K-Nearest Neighbors Support Vector Machines (KNN-SVM), and Long Short- Term Memory Networks (LSTM), have been employed for intrusion detection, each yielding different levels of accuracy. Consequently, the selection of the most suitable model for intrusion detection hinges on a myriad of factors, such as the available computational resources, the characteristics of the network traffic, and the desired level of detection accuracy.

Table 7. Comparison of intrusion detection models using the Cicids2017 dataset.

Author	Model	Accuracy	Dataset
Proposed	DT	0.99	CICIDS2017
Panda et al. [33]	Gradient Boosting	0.9605	CICIDS2017
Kebande et al. [34]	CNN	0.9832	CICIDS2017
Alazab et al. [35]	RF	0.9406	CICIDS2017
Singh et al. [36]	RF	0.9542	CICIDS2017
Tripathi et al. [37]	SVM	0.9202	CICIDS2017

5. Conclusion

In this paper, we proposes a machine learning-based approach for Network intrusion detection in IoD infrastructure and classification, utilizing a set of seven classifiers including DT, RF, Naïve Bayes, ADA, XGB, KNN, and logistic regression. The CICIDS2017 dataset was utilized for evaluating the accuracy of the proposed model after preprocessing. According to the results, the decision tree classifier achieved the highest accuracy rate of 0.99, while the Naïve Bayes classifier achieved the lowest accuracy rate of 0.85. The DT classifier outperformed the other classifiers due to its parametric function evaluation and lower misclassification error. The results of this study suggest that the DT classifier is a suitable choice for NID and classification.

References

1. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
2. R. Hatterjee, R. Chakraborty, and J. K. Mandal, "Design of crypto- graphic model for end-to-end encryption in fpga based systems," in 2019 3rd. IEEE, 2019.
3. G. Choudhary, V. Sharma, T. Gupta, J. Kim, and I. You, "Internet of drones (iod): Threats, vulnerability, and security perspectives," arXiv preprint arXiv:1808.00203, 2018.
4. J. J. Martinez-Montiel, R. L. Sanchez-Villeda, and G. E. Sanchez-Perez, "Intrusion detection system for uavs using machine learning techniques," in 2021 IEEE 7th Colombian Conference on Automatic Control (CCAC). IEEE, 2021, pp. 1–6.
5. HMS Badar, S Qadri, S Shamshad, MF Ayub, K Mahmood, N Kumar, "An identity based authentication protocol for smart grid environment using physical uncloneable function," IEEE Transactions on Smart Grid, vol. 12, no. 5, pp. 4426-4434, 2021.
6. A Akram, R Jiadong, T Rizwan, M Irshad, SM Noman, J Arshad, SU Badar, "A pilot study on survivability of networking based on the mobile communication agents," International Journal of Network Security, vol. 23, no. 2, pp. 220-228, 2021.
7. HMS Badar, K Mahmood, W Akram, Z Ghaffar, M Umar, AK Das, "Secure authentication protocol for home area network in smart grid-based smart cities," Computers and Electrical Engineering, vol. 108, p. 108721, 2023.
8. HMS Badar, MS Obaidat, K Mahmood, N Saher, MF Ayub, DM Khan, "An access control protocol for IoT-based critical infrastructure in smart grid environment," International Journal of Communication Systems, vol. 35, no. 8, p. e5115, 2022.
9. G Naqvi, Z Anwar, AA Khan, A Ahmad, HMS Badar, NI Kajla, "Advancements in Facial Expression-Based Automatic Emotion Identification Using Deep Learning," Journal of Computing & Biomedical Informatics, vol. 5, no. 01, pp. 165-173, 2023.
10. D Rehman, MA Jamil, HMS Badar, MDA Awan, MU Chaudhry, NI Kajla, "Enhancing Crop Production and Water Conservation through IoT-Based Smart Irrigation Systems," Journal of Computing & Biomedical Informatics, vol. 5, no. 01, pp. 96-104, 2023.
11. HI Younas, S Bukhari, F Bukhari, N Aslam, HMS Badar, NI Kajla, "An Efficient Methodology for the Classification of Invasive Ductal Carcinoma Using Transfer Learning," Journal of Computing & Biomedical Informatics, vol. 4, no. 01, pp. 236-250.
12. Z. Abou El Houda, A. Hafid, and L. Khoukhi, "Blockchain meets ami: Towards secure advanced metering infrastructures," in ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020, pp. 1–6.
13. C. D. McDermott and A. Petrovski, "Investigation of computational intelligence techniques for intrusion detection in wireless sensor net- works." International journal of computer networks and communica- tions, vol. 9, no. 4, 2017.
14. H. M. S. Badar, K. Mahmood, W. Akram, Z. Ghaffar, M. Umar, and A. K. Das, "Secure authentication protocol for home area network in smart grid-based smart cities," Computers and Electrical Engineering, vol. 108, p. 108721, 2023.
15. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
16. Z. Abou El Houda, A. S. Hafid, and L. Khoukhi, "A novel machine learning framework for advanced attack detection using sdn," in 2021 IEEE Global Communications Conference (GLOBECOM). IEEE, 2021, pp. 1–6.
17. M. Gupta, "Hybrid intrusion detection system: Technology and develop- ment," International Journal of Computer Applications, vol. 115, no. 9, pp. 5–8, 2015.
18. A. P. Singh and M. D. Singh, "Analysis of host-based and network-based intrusion detection system," International Journal of Computer Network and Information Security, vol. 6, no. 8, pp. 41–47, 2014.
19. T. Micro, "Addressing big data security challenges: The right tools for smart protection," US: Trend Micro, 2012.
20. E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," arXiv preprint arXiv:1701.02145, 2017.
21. G. C. Cawley and N. L. Talbot, "Fast exact leave-one-out cross- validation of sparse least-squares support vector machines," Neural networks, vol. 17, no. 10, pp. 1467–1475, 2004.
22. A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commu- nications surveys & tutorials, vol. 18, no. 2, pp. 1153–1176, 2015.
23. [16]W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of svm and ann for intrusion detection," Computers & Operations Research, vol. 32, no. 10, pp. 2617–2634, 2005.
24. R. A. Francis, S. D. Guikema, and L. Henneman, "Bayesian belief networks for predicting drinking water distribution system pipe breaks," Reliability Engineering & System Safety, vol. 130, pp. 1–11, 2014.
25. G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the cicids2017 case study," in 2021 IEEE Security and Privacy Workshops (SPW). IEEE, 2021, pp. 7–12.
26. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." ICISSp, vol. 1, pp. 108–116, 2018.
27. D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, R. Budiarto et al., "Cicids- 2017 dataset feature analysis with information gain for anomaly detec- tion," IEEE Access, vol. 8, pp. 132 911–132 921, 2020.
28. S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," Expert Systems with applications, vol. 39, no. 1, pp. 129–141, 2012.
29. J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 5, pp. 649– 659, 2008.

30. C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," expert systems with applications, vol. 36, no. 10, pp. 11 994–12 000, 2009.
31. M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001. IEEE, 2000, pp. 38–49.
32. K.-W. Hsu, "Heterogeneous adaboost with stochastic algorithm selec- tion," in Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, 2017, pp. 1–8.
33. S. Panda, B. Majhi, and S. K. Ghosh, "Effective machine learning algorithms for network intrusion detection system: a comprehensive study," Neural Computing and Applications, vol. 32, no. 20, pp. 15 175–15 193, 2020.
34. V. R. Kebande and Y. Zhang, "Convolutional neural network for in- trusion detection in iot networks," Journal of Ambient Intelligence and Humanized Computing, pp. 1–14, 2021.
35. M. Alazab, M. L. M. Kiah, A. A. Ahmed, and K.-K. R. Choo, "Effi- cient and effective intrusion detection model for sdn-based networks," Computers & Security, vol. 84, pp. 1–14, 2019.
36. D. Singh and D. Patel, "Machine learning techniques for intrusion detection: A survey," Journal of Network and Computer Applications, vol. 131, pp. 1–25, 2019.
37. P. Tripathi and S. Misra, "Intrusion detection system using a new hybrid feature selection method and soft computing techniques," The Journal of Supercomputing, vol. 76, no. 2, pp. 1277–1304, 20g, vol. 76, no. 2, pp. 1277–1304, 2020.