University of Aden Journal of Natural and Applied Sciences., vol.28(2) (2024) 53-60

University of Aden Journal of Natural and Applied Sciences

Journal homepage: https://uajnas.adenuniv.com

Research Article

Comparative Analysis of Machine Learning Techniques for Intrusion Detection in IoT Networks

Alaa Mohammed Ali Abdullah* and Khaled Ahmed Abood**

Computer Science and Engineering Department, Faculty of Engineering, University of Aden, Yemen

https://doi.org/10.47372/uajnas.2024.n2.a05

ARTICLE INFO

Received: 28 Oct 2024

Accepted: 14 Feb 2025

Keywords:

IoT, IDS, ML, RF, KNN, NB, XG Boost, SVM, LDA, NSL-KDD

Abstract

The rapid increase of Internet of Things (IoT) devices has introduced significant security challenges, requiring the development of effective Intrusion Detection Systems (IDS) to protect networks from malicious attacks. This study presents a comparative analysis of five machine learning (ML) algorithms (Random Forest (RF), K-Nearest Neighbors (KNN), Naïve Bayes (NB), XGBoost, and Support Vector Machine (SVM)) for IoT intrusion detection using the NSL-KDD dataset. Linear Discriminant Analysis (LDA) is used as a feature extraction technique to optimize model performance by reducing data dimensionality while retaining critical information. Three LDA scenarios with 2, 3, and 4 extracted features are used to compare the mentioned ML algorithms using the performance metrics like accuracy, precision, recall, F1-score, and execution time. The results show that RF achieved the highest accuracy (98.76%) with a slightly higher execution times making it ideal for applications prioritizing accuracy. KNN and XGBoost displayed a balance between high accuracy and computational efficiency, with execution times suitable for real-time IoT applications, with KNN achieving the shortest execution time. The results also highlight the importance of selecting ML algorithms based on the trade-offs between accuracy and efficiency for IoT intrusion detection.

1. Introduction

The fundamental idea behind IoT is to link multitude physical devices via a network, enabling them to efficiently collect and exchange data for intelligent control and management purposes. IoT applications have been widely used in various industries and areas of daily life. In the home, IoT technology facilitates the connection and interaction of devices like lighting, heating, air conditioning, televisions, and refrigerators over a network, enabling functions such as remote control, automated operations, and more. IoT also can monitor industrial production processes to enhance productivity and reduce operational costs. In city management, IoT can be used for traffic control, environmental monitoring, and enhancing public safety measures. Furthermore, IoT has various applications in healthcare, agriculture, and marketing sectors [1].

IDS is one among the most powerful dynamic mechanism that determines and detects the specific attacks in the network by using set of actions that attempts to compromise the integrity, confidentiality, or availability of any resource by monitoring and analysing the process of network. Artificial Intelligence and machine learning-powered IDS solutions have been increasingly employed in IoT environments. These advanced systems are capable of autonomously learning and recognizing typical network behaviour patterns, enabling the efficient detection of abnormal activities. IDSs serve to protect IoT devices from attacks by detecting intrusions and notifying about anomalous behaviours, preempting intruders from breaching the IoT network [2], Figure.1 illustrates the implementation of an IDS within an IoT environment, displaying the deployment of IoT devices and servers on the public Internet.

In this research we present a Machine Learning (ML) based IDS that uses the classification ML Algorithms. ML is a

*Correspondence to: Computer Science and Engineering Department, Faculty of Engineering, University of Aden, Yemen
E-mail address: aloosh200@gmail.com

technique that gives computer programs the ability to learn tasks without being explicitly instructed to do so. ML based systems are capable of learning from experience [3].
Four key machine learning classification algorithms are applied in this research.

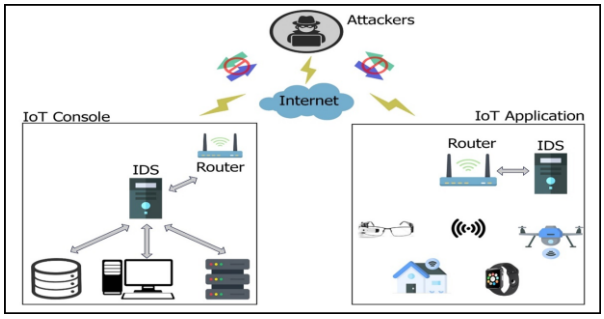


Fig : Figure 1. An IDS IoT environment

The model's performance is analysed and evaluated. The subsequent sections are structured as follows: Section 2 provides background related to our study. Section 3 outlines the methodologies, encompassing dataset selection, preprocessing, feature selection [4], and classification techniques. Section 4 shows experimental setup. Section 5 details the results, while Section 6 offers the study's conclusion.

1.1. Related Work:

There are several cybersecurity-related challenges in the IoT environment. One of the workable solutions refers to detect attacks in IoT using Machine Learning. M. Anwer et al. [5] recommended a framework for the detection of malicious network traffic. The framework uses three popular classification-based malicious network traffic detection methods, namely Support Vector Machine (SVM), Gradient Boosted Decision Trees (GBDT), and Random Forest (RF). While achieving an accuracy rate of 85.34% with (RF).
M. Nasser et al. [6] studied an IDS model to reduce the NSL-KDD dataset features using ANOVA-PCA and compared with other feature selection algorithms such as neighbor component analysis (NCA) and ReliefF and displayed good classification accuracy. E. Gbashi et al. [7] suggested model that gives satisfactory results of accuracy rate using NSL-KDD and they suggested for enhancement as future work to use LDA for features extraction.Since accuracy alone cannot be considered as satisfactory metric, in this work our intention is to evaluate the execution time and to use analysis technique presented by R. Dattiet al. [8] to use LDA for transforming dataset features to achieve optimal performance levels.

2. Methodology

In this section we explain the experimental method carried out through the research study as shown in Figure 2

2.1. Data Collection

The NSL-KDD dataset [9] is an improved version of the KDD'99 dataset, which is widely used for intrusion detection research in IoT. The KDD'99 dataset was created for the DARPA Intrusion Detection Evaluation Program and contains a large number of network connection records. However, the KDD'99 dataset has some limitations, such as redundancy and irrelevant features. The NSL-KDD dataset was proposed to address these issues by removing redundant records and features from the KDD'99 dataset. This makes the NSL-KDD dataset more suitable for intrusion detection research as it reduces the complexity and processing time required for analysis [10].

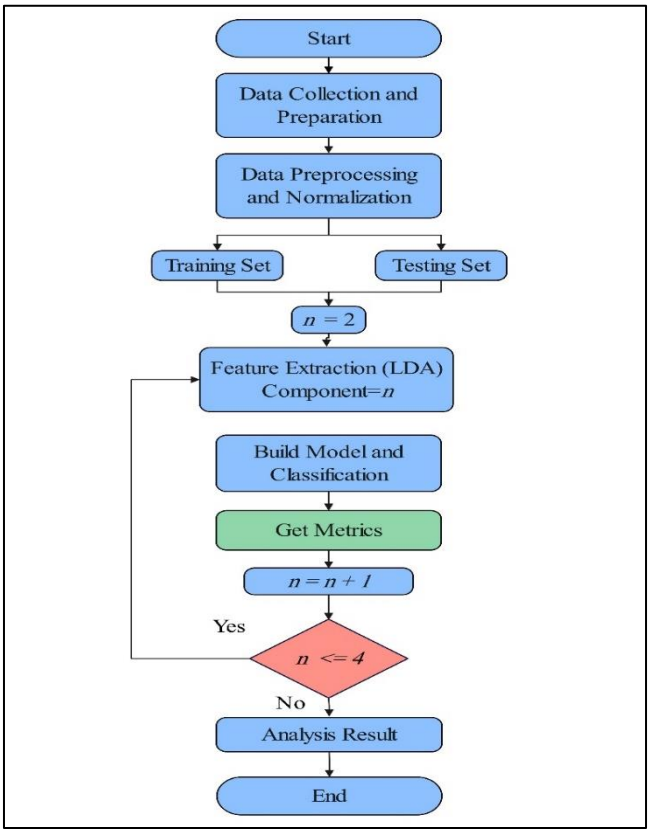


Figure2. Proposed system

NSL-KDD dataset consists of chosen records of all KDD data set. The training dataset consists of (125,973) and test dataset consists of (22,544) samples each sample contains 42 features as shown on Table.1.
Attacks in this dataset [11] are split into four types:

1. Denial of Service (DoS): DoS attacks have the objective of blocking or restricting services delivered by the network, computer to the users – e.g. syn flooding.

2. Probing: probing attacks have the objective of acquisition of information about the network or the computer system – e.g. port scanning.
3. Remote to Local (R2L): R2L describe as unauthorized access from a remote machine, the attacker intrudes into a remote machine and gains local access of the victim machine – e.g. password guessing.
4. User to Root (U2R): Unauthorized access to local super user (root) privileges is an attack type, by which an attacker uses a normal account to login into a victim system and tries to gain root/administrator privileges by exploiting some vulnerability in the victim – e.g. buffer overflow attacks.

2.2. Data Preparation

Data preparation is the crucial initial stage where raw data is meticulously cleaned and transformed before undergoing further processing and analysis. This essential process involves rectifying errors, restructuring data formats. The primary phase of data preparation involves segregating connections into normal and anomaly classes based on the 'target' column. Subsequently, attacks are categorized into four primary classes: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L), as illustrated in Figure 3.

Feature Name	Feature Name	Feature Name
duration	su_attempted	same_srv_rate
protocol_type	num_root	diff_srv_rate
service	num_file_creations	srv_diff_host_rate
flag	num_shells	dst_host_count
src_bytes	num_access_files	dst_host_srv_count
dst_bytes	num_outbound_cmds	dst_host_same_srv_rate
land	is_host_login	dst_host_diff_srv_rate
wrong_fragment	is_guest_login	dst_host_same_src_port_rate
urgent	count	dst_host_srv_diff_host_rate
hot	srv_count	dst_host_serror_rate
num_failed_logins	serror_rate	dst_host_srv_serror_rate
logged_in	srv_serror_rate	dst_host_rerror_rate
num_compromised	rerror_rate	dst_host_srv_rerror_rate
root_shell	srv_rerror_rate	target

2.3. Data Pre-processing

Data preprocessing is essential in Machine Learning, affecting how effectively models learn. Key tasks include managing null values, standardization, handling categorical variables, discretization, and encoding [12]. Real-world data is often incomplete and error-prone, requiring preprocessing to clean and enhance it. Techniques like One-Hot Encoding used to encode categorical values, the 42 features in the dataset became 122 features. Then, MinMax Scalar used to scale numerical features to improve utilization of machine learning algorithms.

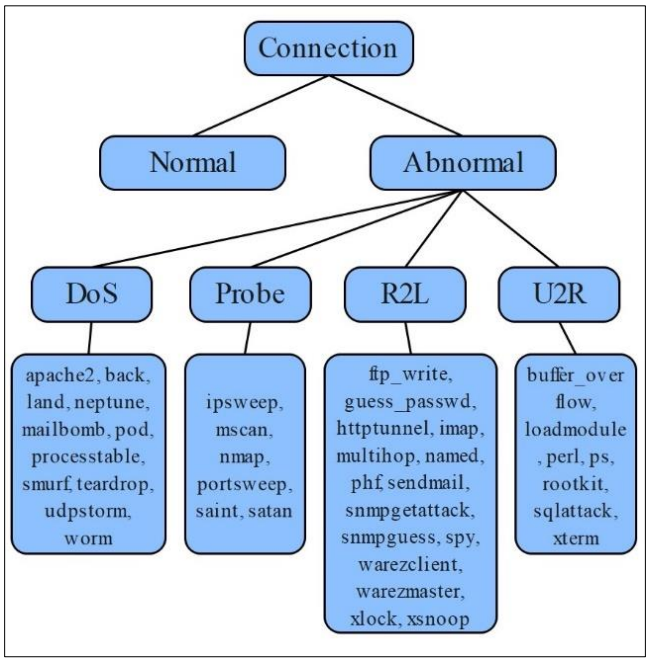


Figure.3: Classification of intrusion attacks in NSL-KDD dataset

2.4. Feature Extraction

Feature extraction is a crucial step in machine learning where raw data is transformed into relevant features to simplify model training and analysis [13]. By reducing data dimensionality while retaining essential information, this process improves model performance, accelerates computation, and enhances predictive accuracy. LDA technique [14] used as a dimensionality reduction in the preprocessing step for better model optimization and predictive outcomes. The model iterates LDA to transformed and extract 2, 3, and 4 features at a time to feed machine learning algorithm with the extracted features, evaluating all metrics in each scenario.

2.5. Classification Algorithms:

1. **Random Forest (RF):** it is a versatile ensemble learning algorithm that constructs multiple decision trees during training and combines their outputs to make predictions. Known for its accuracy and robustness against overfitting, Random Forest is widely used for classification tasks, especially with complex datasets [15].
2. **Naïve Bayes (NB):** Naive Bayes is a probabilistic ML algorithm relying on Bayes' theorem with a naive assumption of feature independence. known for simplicity and efficiency in classification, NB calculates class probabilities from input features, making it ideal for tasks like text classification and spam filtering, where its assumptions match data characteristics [15].

3. **Support Vector Machine (SVM):** SVMs are ML algorithms that attempt to use hyperplane-based vectors in order to separate the data's labels with maximal margin. With this hyperplane separation, the SVM is then able to "understand" the spatial location of the classes within the dataset [16].
4. **K-Nearest Neighbours (KNN):** it is a simple yet effective algorithm for classification and regression tasks. It assigns labels to new data points based on the majority class of their neighbouring points, making it easy to implement and understand, particularly in scenarios where data is non-linear [16].
5. **XGBoost:** it is a powerful boosting algorithm known for its efficiency and performance in classification and regression tasks. It utilizes gradient boosting techniques to enhance accuracy and speed, making it a popular choice in various machine learning applications [17].

2.6. Evaluation Metrics:

Evaluation metrics are essential in assessing the performance of Machine Learning models used for classifying Intrusion Detection Systems (IDS), Our study computed the confusion matrix values as shown in Table.2 to evaluate performance metrics in addition to execution time.

Table.2: Confusion Matrix sections

True Positive (TP)	The model correctly predicted normal as normal.
True Negative (TN)	The model correctly predicted attacker as attacker
False Positive (FP)	The model incorrectly identifies a normal activity as a malicious one
False Negative (FN)	The model incorrectly identifies malicious traffic as normal

The model computed the following performance metrics:

1. Accuracy (**A**): The proportion of correctly classified instances.

$$\mathbf{A} = \frac{TP + TN}{Total\ Number\ of\ Samples} * 100$$

2. Precision (P): The proportion of positive identifications was actually correct.

$$\mathbf{P} = \frac{TP}{TP + FP}$$

3. Recall **(R):** Recall or sensitivity is a measure that tells us what proportion of actual positives was identified correctly.

$$\mathbf{R} = \frac{TP}{TP + FN}$$

4. F1 Score (F1): The measure that combines both precision and recall into a single value. It is the harmonic mean of precision and recall.

$$\mathbf{F1} = 2 * \frac{P + R}{P * R}$$

5. Training Time **(TT):** The duration it takes for a machine learning model to learn patterns and relationships within a dataset during the training phase.
6. Prediction Time **(PT):** The duration taken by the trained model to generate predictions.
7. Execution Time **(ET):** The total time spent for both training time **(TT)** and predictions time **(PT). ET = TT + PT**

2.7.Experimental Setup

The research is carried out using a DELL Latitude system running the Windows 10 Enterprise OS, powered by an i7-8650U Processor, 32GB of memory, and an integrated Intel UHD 620 graphics card. Numpy and Pandas libraries in Python [18] were used for tasks such as data preprocessing, cleaning, and feature selection.

3. Result Analysis and Discussion:

The evaluation of five machine learning algorithms (Random Forest (RF), K-Nearest Neighbors (KNN), Naive Bayes, XGBoost, and SVM) was conducted under three feature extraction scenarios using Linear Discriminant Analysis (LDA) as dimensional reduction technique on the NSL-KDD dataset. Each model evaluated using metrics such as accuracy, precision, recall, F1-score, and execution time. The results are summarized in Table.3.

Table.3: Experimental Results

LDA Scenarios	ML Model	Random Forest	KNN	Naive Bayes	XGBoost	SVM
Scenario No.1 (2 Features extracted)	Accuracy	96.61	96.22	93.04	96.03	93.66
	Train Time	29.092	0.078	0.031	2.707	188.998
	Predict Time	0.594	1	0.016	0.094	61.695
	Execution Time	29.686	1.078	0.047	2.801	250.694
	Precision	96.46	96.12	93.33	95.92	92.65
	Recall	96.61	96.22	93.04	96.03	93.66
	F1-Score	96.51	96.16	91.79	95.9	92.37
Scenario No.2 (3 Features extracted)	Accuracy	98.29	98.04	93.59	97.94	94.78
	Train Time	26.522	0.109	0.031	2.908	210.289
	Predict Time	0.532	1.141	0.016	0.087	53.997
	Execution Time	27.053	1.251	0.047	2.995	264.287
	Precision	98.27	98	94.03	97.9	94.8
	Recall	98.29	98.04	93.59	97.94	94.78
	F1-Score	98.25	98.01	93.78	97.91	94.41
Scenario No.3 4 Features extracted	Accuracy	98.76	98.45	92.99	98.54	95.6
	Train Time	50.282	0.125	0.031	3.066	106.945
	Predict Time	0.47	1.242	0.016	0.094	52.829
	Execution Time	50.752	1.367	0.047	3.159	159.773
	Precision	98.75	98.44	93.48	98.52	95.57
	Recall	98.76	98.45	92.99	98.54	95.6
	F1-Score	98.75	98.44	93.2	98.52	95.48

3.1- Result Discussion:

3.1.1- Scenario No. 1: Two Features Extracted

In the first scenario, where two features were extracted:

- **Accuracy**: Random Forest achieved the highest accuracy (96.61%), followed by KNN (96.22%) and XGBoost (96.03%). Naive Bayes and SVM performed lower at 93.04% and 93.66%, respectively.
- **Execution Time**: Naive Bayes exhibited the fastest execution time (0.047 seconds), while SVM showed the slowest (250.694 seconds). Random Forest (29.686 seconds), KNN (1.078 seconds), and XGBoost (2.801 seconds) showed moderate execution times.
- **Precision, Recall, and F1-Score**: Random Forest and KNN displayed similar performance across these metrics, with Random Forest slightly better. Naive Bayes had the lowest F1-Score (91.79%), whereas XGBoost and SVM were balanced but lower than Random Forest and KNN.

3.1.2- Scenario No. 2: Three Features Extracted

With three features extracted, the results improved for most models:

- **Accuracy**: Random Forest outperformed others with an accuracy of 98.29%, followed by KNN (98.04%) and XGBoost (97.94%). Naive Bayes (93.59%) and SVM (94.78%) remained lower.
- **Execution Time**: Naive Bayes remained the fastest (0.047 seconds), while SVM had the longest execution time (264.287 seconds). KNN provided a low execution time (1.251 seconds), making it efficient for real-time applications.
- **Precision, Recall, and F1-Score**: Random Forest and KNN achieved almost equal performance with high precision (98.27% and 98%, respectively) and F1-Scores (98.25% and 98.01%). XGBoost remained competitive, while Naive Bayes and SVM trailed.

Figure.4: ML Accuracy for each scenario

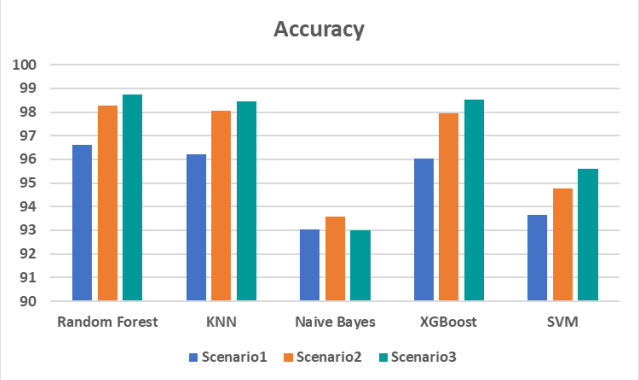
3.1.3- Scenario No. 3: Four Features Extracted

The third scenario, involving four features, presented the best overall accuracy but varied execution times:

- **Accuracy**: Random Forest achieved the highest accuracy (98.76%), followed by XGBoost (98.54%) and KNN (98.45%). SVM (95.6%) and Naive Bayes (92.99%) scored lower, as shown in Figure.4.
- **Execution Time**: Naive Bayes kept its efficiency with a minimal execution time (0.047 seconds), while Random Forest required 50.752 seconds and SVM 159.773 seconds. KNN and XGBoost showed reasonable times, making them suitable for real-time applications.
- **Precision, Recall, and F1-Score**: Random Forest achieved the best scores (98.75% for all three metrics), followed by XGBoost and KNN. Naive Bayes and SVM had comparatively lower scores, indicating limitations when features increased.

3.2- Impact of Feature Extraction (LDA)

The application of Linear Discriminant Analysis (LDA) for feature extraction had a significant impact on the performance of the machine learning models. We observed that, as the number of features increased from 2 to 4, all models displayed improved accuracy and F1-Scores. This indicates that LDA effectively retained critical information while reducing dimensionality, leading to better model performance. On the other hand, increasing number features increased execution times, particularly for computationally intensive models like Random Forest and SVM. While KNN and Naive Bayes showed low execution times, even with more features, making them suitable for real-time IoT applications. This prove significance of using LDA as a preprocessing step for Intrusion Detection Systems in IoT networks, where computational resources are limited.



4. Conclusion and Future Work

Securing IoT devices from malicious attacks requires robust Intrusion Detection Systems (IDS), and machine learning (ML) algorithms have shown significant potential in enhancing the accuracy and efficiency of these systems. This study presents a comparative analysis of five ML techniques (Random Forest (RF), K-Nearest Neighbors (KNN), Naïve Bayes, XGBoost and SVM) for intrusion detection in IoT networks using the NSL-KDD dataset. With Linear Discriminant Analysis (LDA) applied for feature extraction, the analysis highlights the importance of selecting machine learning models based on specific IoT requirements, such as the trade-off between accuracy and execution time. Random Forest is suitable for applications prioritizing accuracy, while KNN and XGBoost are better choices for real-time scenarios. Naive Bayes and SVM, while having limitations, can still find relevance in specialized applications where these limitations can be effectively addressed. Future work should focus on optimizing these models, exploring their performance with other IoT datasets, and developing more robust and effective IDS solutions adapted to IoT security needs.

.

References

1. P. Aggarwala, S.Sharmab. Analysis of KDD Dataset Attributes - Class wise For Intrusion, International Conference on Recent Trends in Computing (ICRTC). Vol. 57, 2015, 842 – 851.
2. H. Liao, M. Z. Murah, M. K. Hasan, A. M. Aman, J. Fang, X. Hu, AND A. U. Rehman Khan. A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things, IEEE Access, Vol. 12, 2017, 4745 - 4761.
3. S M Kasongo. A deep learning technique for intrusion detection system using a Recurrent, Department of Industrial Engineering & The School for Data Science and Computational Thinking, Stellenbosch University, Vol. 199, 2023, 113-125.
4. R. Rahim, A. S Ahanger, S. M Khan and F. Masoodi, Analysis of IDS using Feature Selection Approach on NSL-KDD Dataset, in SCRS Conference Proceedings on Intelligent Systems, 475–481, Computing & Intelligent Systems, SCRS, India, 2021.
5. M. Anwer, S.H. Khan, M.U. Farooq and Waseemullah. Attack Detection in IoT using Machine Learning. Engineering, Technology & Applied Science Research. Vol. 11, No. 3, 2021, 7273-7278.
6. M. Nasser, M. Ahmed. Data Preparation and Reduction Technique in Intrusion Detection Systems: ANOVA-PCA, International Journal of Computer Science and Security (IJCSS). Vol.13, 2019, 167 – 220.
7. E. Gbashi, B. Mohammed. Intrusion Detection System for NSL-KDD Dataset Based on Deep Learning and Recursive Feature

Elimination, Engineering and Technology Journal, Vol. 39, 2021, 1069 – 1079.

8. R. Datti, B. verma. Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis, International Journal on Computer Science and Engineering (IJCSE), Vol. 02, 2010, 1072-1078.
9. NSL-KDD research dataset, Canadian Institute for Cybersecurity.
10. R. R. Devi and M. Abualkibash, Intrusion detection system classification using differently machine learning algorithms on KDD-99 and NSL-KDD datasets. International Journal of Computer Science & Information Technology, Vol 11, No 3, 2019, 11306.
11. L. Dhanabal, and P. Shantharajah, A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, 2015, 446-452.
12. S. Revathi and A. Malathi, A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection, International Journal of Engineering Research & Technology (IJERT), Vol. 2, Dec 2013, 2278-0181.
13. M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, M. Portmann. Feature extraction for machine learning-based intrusion detection in IoT networks, Digital Communications and Networks, Vol.10, 2024, 205-216.
14. Q. Wu, X. Zhao, A linear discriminant analysis-based algorithm for identifying anomalous traffic in large-scale networks, Applied Mathematics and Nonlinear Sciences, 9, 2024 1-17.
15. S. Sapre and P. Ahmadi, K. Islam. A Robust Comparison of the KDDCup99 and NSL-KDD IoT Network Intrusion Detection Datasets Through Various Machine Learning Algorithms, Department of Information Sciences and Technology George Mason University, Vol 01, 2019, 1912.13204.
16. R. N. Rithesh, SVM-KNN: A Novel Approach to classification Based on SVM and KNN, International Research Journal of Computer Science (IRJCS), Vol 4, August 2017.
17. Dhaliwal SS, Nahid A-A, Abbas R. Effective Intrusion Detection System Using XGBoost. Information. 2018; 9(7)149.
18. Scikit-Learn User Guide (Python), Release 0.21. A. Dalmia, L. Esteve. https://scikit-learn.org/0.21/_downloads/scikit-learn-docs.pdf.

University of Aden Journal of Natural and Applied Sciences., vol.28(2) (2024) 53-60

مجلة جامعة عدن للعلوم الطبيعية والتطبيقية

Journal homepage: https://uajnas.adenuniv.com

بحث علمي

تحليل مقارنة لتقنيات التعلم الآلي للكشف عن الاختراق في شبكات إنترنت الأشياء

علاء محمد علي عبدالله و خالد أحمد عبود

قسم علوم و هندسة الكمبيوتر

كلية الهندسة ، جامعة عدن

https://doi.org/10.47372/uajnas.2024.n2.a05

الملخص

أدى الارتفاع السريع في أجهزة إنترنت الأشياء (IoT) إلى ظهور تحديات أمنية كبيرة، مما يتطلب تطوير أنظمة فعالة لكشف التطفل (IDS) لحماية الشبكات من الهجمات الضارة. تقدم هذه الدراسة تحليلاً مقارنًا لخمس خوارزميات للتعلم الآلي (ML) (الغابة العشوائية (RF)، وأقرب جيران K (KNN)، وخوارزمية بايز الساذجة (NB)، وخوارزمية XGBoost، وآلة دعم المتجهات (SVM)) للكشف عن التطفل في إنترنت الأشياء باستخدام مجموعة بيانات NSL-KDD. يتم استخدام تحليل التمييز الخطي (LDA) كتقنية لاستخراج الميزات لتحسين أداء النموذج من خلال تقليل أبعاد البيانات مع الاحتفاظ بالمعلومات الهامة. يتم استخدام ثلاثة سيناريوهات لتحليل التمييز الخطي مع 2 و3 و4 ميزات مستخرجة لمقارنة خوارزميات التعلم الآلي المذكورة باستخدام مقاييس الأداء مثل الدقة والدقة والاستدعاء ودرجة F1 ووقت التنفيذ. تظهر النتائج أن RF حقق أعلى دقة (98.76٪) مع أوقات تنفيذ أعلى قليلاً مما يجعله مثاليًا للتطبيقات التي تعطي الأولوية للدقة. أظهرت KNN وXGBoost توازنًا بين الدقة العالية والكفاءة الحسابية، مع أوقات تنفيذ مناسبة لتطبيقات إنترنت الأشياء في الوقت الفعلي، حيث حققت KNN أقصر وقت تنفيذ. تسلط النتائج أيضًا الضوء على أهمية اختيار خوارزميات التعلم الآلي بناءً على المقايضات بين الدقة والكفاءة لاكتشاف اقتحام إنترنت الأشياء.

مفاتيح البحث

التسليم : 28 أكتوبر 2024

القبول :14 فبراير 2025

كلمات مفتاحية :

IDS، LDAأنظمة كشف التطفل، ، إنترنت الأشياء، التعلم NSL-KDD ، AdaBoost، KNN، RFالآلي، XGBoostح.