A Comparative Study of Machine Learning Algorithms Used for Network Intrusion Detection

ADEBANJO ADESHINA WASIU[1], IJEGWA DAVID ACHEME[2]
[1]*Department of ICT Education, Africa Center of Excellence for Innovative and Transformation STEM Education (ACEITSE), Lagos State University, Lagos, Nigeria*
[2]*Department of Computer Science, Edo State University, Uzairue, Nigeria*

Abstract- ***Across different sectors of human endeavors such as health, aviation, agriculture, education, and finance, institutions and organizations are increasingly embracing ICT infrastructures, relying more on computers and cyber resources for their daily operations. However, this growing dependence on cyber systems has led to a corresponding rise in cyber-attacks. Therefore, there's a pressing need to develop robust countermeasures to safeguard confidential information and ensure its availability. Among the many techniques employed by attackers to breach computer network security, intrusion stands out as one of the most common significant attack type. Numerous research endeavors have been dedicated to developing intrusion detection systems (IDS) to address this challenge. The focus of this study is the exploration of selected machine learning techniques that have been reported for IDS development. To accomplish this, the research builds a predictive machine learning model using four popular algorithms (Logistic Regression, Random Forest, Decision trees, and Naïve Bayes) for the detection and prediction of suspicious connections. This was achieved through the analysis of the KDD Cup 1999 dataset, wherein machine learning algorithms are employed to identify patterns and anomalies, which can enable business owners to deploy preemptive measures against potential security breaches. Subsequently, the performances of these algorithms are evaluated and ranked based on their prediction accuracy and other established performance metrics. The results in terms of prediction accuracy show that the Random Forest algorithm performed best, followed by the decision tree, then Logistic regression and finally, naïve bayes with the least accuracy.***

Indexed Terms- ***Classification Algorithms, Cybersecurity, Intrusion Detection Systems***

I. INTRODUCTION

IDS are systems which are mainly designed to protect the integrity, availability, and confidentiality of computer resources that are in a networked environment (Varanasi and Razia, 2022). With organizations and companies increasingly reliant on cloud computing and cyber-based systems, the necessity for effective cyber security measures has become even more important. Hackers and intruders persistently pose a growing threat, making relentless attempts to compromise networks and web services through unauthorized access. Among the popular cyber security methods is the utilization of intrusion detection systems (IDS), which can be developed using machine learning algorithms. These algorithms are employed to scrutinize and categorize connections as either suspicious or safe (Ahmad et al., 2021). The implementation of machine learning algorithms for intrusion detection and prevention has garnered considerable attention in academic research. These investigations have delved into the predictive capabilities concerning the four core classifications of cyber attacks, namely: denial-of-service (DoS) attacks, malware threats, insider threats, and unauthorized access attempts. Such efforts aim to improve cybersecurity measures by leveraging advanced computational techniques for preemptive threat identification and mitigation. Other attack types include;

1) Access to remote machines without authorization (R2L): for example, password guessing;
2) Access to local root previledges (U2R) for example “buffer overflow” attacks
3) Probing for example port scanning.

Cybersecurity remains an ever-increasing challenge, with the major challenges revolving around the confidentiality, integrity, and availability of computer resources (Bakhsh et al., 2019). These aspects are

integral in ensuring the security of cyber-based systems. Given the critical role cybersecurity plays, numerous research endeavors have emerged, presenting diverse techniques to combat it. More recently, there has been a surge in research focusing on the utilization of machine learning methods for intrusion prediction. This research article aims to implement and evaluate various machine learning (ML) algorithms used for this purpose.

According to Manavalan et al. (2018), the reported machine learning algorithms exhibit varying levels of prediction accuracy and efficiency. This study proposes to implement, compare, and rank the performances of these algorithms for predicting intrusions in networks. By doing so, it seeks to identify the most effective ML algorithm suited to network scenarios and recommend its adoption. Consequently, the findings of this research will establish a framework for the application of more efficient machine learning models in real-time network intrusion prediction (Acheme, & Vincent 2021, Makinde & Acheme, 2023).

A. Research Questions

The primary objective of this study is to ascertain the most effective machine learning algorithm or model for predicting suspicious connections/intrusions within a network, in order to achieve this, we leveraged on the established performance metrics as delineated in Hossin and Sulaiman (2015). These metrics include; prediction accuracy, Precision, Recall, R2 Score, and Confusion Matrix. The research work will therefore be guided by the following specific research questions:

1) Which machine learning algorithm proves to be the most effective in predicting network intrusions, considering prediction accuracy, error rate, confusion matrix, Precision, and Recall as evaluation metrics?
2) What are the key features essential in constructing a machine learning-based intrusion detection system?

II. REVIEW OF RELATED LITERATURE

While there are various techniques and models for detecting intrusion in computer networks have been documented in the literature, this section reviews a few closely related works that have utilized machine learning algorithms:

Tahri et al. (2022) introduced an intrusion detection system employing three distinct classification algorithms—Naive Bayes Support Vector Machine, K-Nearest Neighbors, and K-Nearest Neighbors. Leveraging the USNW NB 15 DATASET, the chosen algorithms were initially implemented and evaluated. Subsequently, the database was processed using the most effective algorithm determined from the initial assessment. The model's performance was evaluated using two separate datasets, NSL-KDD and UNSW-NB15 in order to assure its effectiveness.

Thomas & Roopam (2021) in their work demonstrated NID-Shield. This was a hybrid network intrusion detection system (IDS) that categorizes datasets based on various attack types. Furthermore, attack names within the attack types were individually categorized, thereby aiding in assessing the vulnerability of each assault across different networks. The hybrid NID-Shield NIDS employed the effective feature subset selection method, R, and its performance metrics were evaluated using NSL-KDD and UNSW-NB15 datasets. Their results showed high accuracy and low false positive rates.

Patgiri et al. (2018) showcased the creation of intrusion detection systems (IDS) utilizing Random Forest and Support Vector Machine. The NSLKDD dataset was utilized for model testing, training, and evaluation, with crucial features selected using the recursive feature elimination (RFE) technique.

Jamadar (2018) presented a model for creating network intrusion detection using the decision trees algorithm, focusing on anomaly-based intrusions. The model was developed following standard procedures for machine learning models creation, with the best features chosen using Recursive Feature-Elimination (RFE) methods and categorical features encoded using label encoders.

Kumar & Doegar (2018) highlighted a machine learning model for IDS, emphasizing the use of entropy filter analysis for feature selection. Three types of machine learning models—Naive Bayes, Adaptive Boost, and Partial Decision Trees—were implemented and their performances compared.

Yadav and Sharma (2021) presented analytical insights from intrusion detection systems currently in operation. Additionally, the study examined and provided valuable datasets, along with evaluating several methods for developing efficient Intrusion Detection Systems (IDS)

employing single, hybrid, and ensemble machine learning algorithms. The literature's methods were scrutinized and compared using diverse datasets to offer clear directions and recommendations for future fruitful research endeavors.

III. BACKGROUND OF SELECTED MACHINE LEARNING ALGORITHMS

A. Gaussian Naïve Bayes

The Naive Bayes algorithm, a probabilistic approach in machine learning, serves as a versatile tool in numerous classification scenarios. It is utilized in many classification tasks such as document categorization, spam detection, and predictive modeling, its name “Bayes” comes from the foundational principles laid by Thomas Bayes, While, "Naive" encapsulates the algorithm's fundamental assumption of feature independence, this means that changes in one feature do not affect others within the model. This inherent simplification makes the algorithm to be both simple and resilient, hence, a popular choice in various applications. Equation 1 is a formal representation of its operational framework. Owing to its simplistic yet effective nature, it has gained much application across diverse domains because of its ability to handle large datasets efficiently. Its reliance on basic probabilistic principles facilitates rapid implementation and interpretation, making it particularly suitable for scenarios where computational resources or training data are limited. Despite its inherent assumption of feature independence, Naive Bayes often yields competitive performance compared to more complex algorithms, especially in situations where the data aligns with its underlying assumptions. Furthermore, its interpretability makes it an attractive choice in contexts where model transparency and explainability are important, such as in regulatory compliance or medical diagnosis. Thus, Naive Bayes stands as an important cornerstone in the field of machine learning research.

$$P(X|Y=c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \frac{-(x-\mu_c)^2}{2\sigma_c^2}$$

……. (1)

As seen from eqn 1, sigma (σ) represents the variance and mu (μ) denotes the mean of the continuous variable X calculated for a specific class c of Y.

B. Decision Tree

Decision Trees (DT) belong to the group of supervised machine learning algorithms, the DT algorithm is a versatile tool proficient that has been applied to both regression and classification tasks. Its primary objective revolves around constructing a training models that can predict the class or value of the target variable by synthesizing straightforward decision rules extracted from historical data (training data). In the context of class label prediction, a DT model will commence its traversal at the root of the tree, where it checks the attribute values of the root against those of the record's attribute. Depending on the outcome of this evaluation, the algorithm progresses along the appropriate branch, moving on to the next node in the process. DTs offer several advantages beyond their simplicity. They are inherently interpretable, which allows stakeholders to understand and trust the decision-making process. Moreover, they handle both numerical and categorical data with ease, making them applicable in a wide range of scenarios, De'ath & Fabricius (2000). However, DTs have been widely reported to be susceptible to overfitting, especially when the tree grows deep or when the dataset is noisy. To mitigate this, techniques such as pruning or using ensemble methods like Random Forests are often employed. In addition to their utility in standalone applications, Decision Trees serve as building blocks for more complex algorithms. For instance, ensemble methods like Gradient Boosting Machines often use shallow Decision Trees as weak learners to boost predictive performance.

C. Random Forest

Random Forest (RF) is another widely used supervised machine learning algorithm which is also proficient in tackling both regression and classification problems. It is an aggregation of DTs and operates by constructing multiple decision trees from varied subsets of the dataset, harnessing their collective decision through majority ‘voting’ for classification tasks and averaging for regression predictions. Importantly, Random Forest exhibits a remarkable capability to manage both continuous and categorical data variables, making it versatile across diverse domains. In classification scenarios, it often outperforms other algorithms, because of its robustness and ability to handle complex datasets with high dimensionality and noisy features. Moreover, Random Forest offers several advantages beyond its predictive prowess. It is resistant to overfitting, owing to

the inherent diversity among the constituent trees. Additionally, it provides insights into feature importance, aiding in understanding the underlying patterns driving predictions. Furthermore, its parallelizable nature makes it suitable for large-scale applications, using parallel processing for efficient computation. Despite its strengths, several research works have also reported its limitations, some of these are; it may not perform optimally on extremely imbalanced datasets, where minority classes are underrepresented, also, its interpretability might be limited compared to simpler models like Decision Trees.

D. Logistic Regression

The Logistic regression algorithm predicts binary outcomes based on independent variables. These outcomes typically represent two possibilities: the presence (1) or absence (0) of an event. Independent variables, also known as features, have the potential to affect the outcome, which is the dependent variable. Logistic regression is widely used in classification tasks, especially when the dependent variable is dichotomous or categorical. Its appeal lies in its ability to provide interpretable results and handle both linear and nonlinear relationships between the independent variables and the outcome. Additionally, logistic regression offers insights into the probability of occurrence for each outcome class, aiding decision-making processes.

IV. DATA COLLECTION

The Knowledge Discovery and Data Mining (KDD) Cup 1999 dataset has been used as a benchmark dataset in many of cybersecurity and machine learning modelling problems. This dataset was used as part of The Third International Knowledge Discovery and Data Mining Tools Competition, held in conjunction with the KDD-99 conference. The dataset was created to address the task of network intrusion detection, specifically distinguishing between "bad" connections (intrusions or attacks) and "good" normal connections in a network environment. It comprises a large number of network connection records, each characterized by numerous features such as protocol type, service, duration, and number of failed login attempts as shown in figure 1. The dataset has been widely used for research and evaluation of intrusion detection algorithms and techniques. And has been utilized in this work to develop a network intrusion detection system capable of distinguishing between "bad" connections (intrusions or attacks) and "good" normal connections. It has forty-three features including the target variable labeled as "attack type," categorized into normal or suspicious classes. Figures 3.1 and 3.2 provide insights into the dataset's structure.

protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate
tcp	http	SF	181	5450	0	0	0	0	...	1.0	0.0
tcp	http	SF	239	486	0	0	0	0	...	1.0	0.0
tcp	http	SF	235	1337	0	0	0	0	...	1.0	0.0
tcp	http	SF	219	1337	0	0	0	0	...	1.0	0.0
tcp	http	SF	217	2032	0	0	0	0	...	1.0	0.0

Figure 3.1: labelled dataset

dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	target	Attack Type
0.11	0.0	0.0	0.0	0.0	0.0	normal.	normal
0.05	0.0	0.0	0.0	0.0	0.0	normal.	normal
0.03	0.0	0.0	0.0	0.0	0.0	normal.	normal
0.03	0.0	0.0	0.0	0.0	0.0	normal.	normal
0.02	0.0	0.0	0.0	0.0	0.0	normal.	normal

Figure 3.2: continuation of 3.1

A. Feature Selection

In order to select the optimal features for modeling, a correlation analysis was conducted to pinpoint the most significant ones, while features with minimal correlation were eliminated. Figure 3.3 illustrates the correlation heatmap, showcasing the degree of correlation among all features. Subsequently, based on the correlation analysis results, the least influential features were excluded, as depicted in Figure 3.4

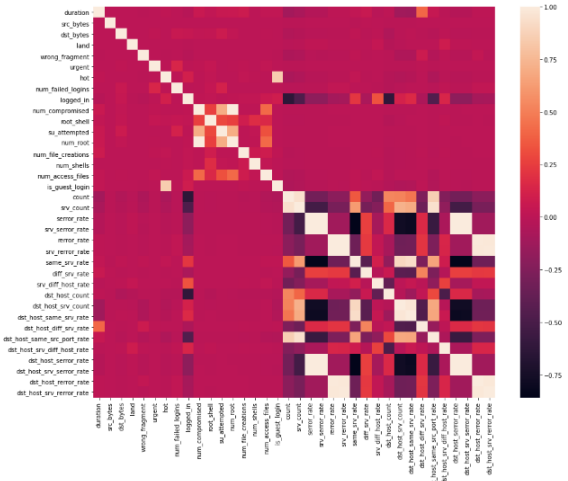

Figure 3.3 correlation of all the features

```
#This variable is highly correlated with num_compromised and should be ignored for analysis.
#(Correlation = 0.9938277978738366)
df.drop('num_root',axis = 1,inplace = True)

#This variable is highly correlated with serror_rate and should be ignored for analysis.
#(Correlation = 0.9983615072725952)
df.drop('srv_serror_rate',axis = 1,inplace = True)

#This variable is highly correlated with rerror_rate and should be ignored for analysis.
#(Correlation = 0.9947309539817937)
df.drop('srv_rerror_rate',axis = 1, inplace=True)

#This variable is highly correlated with srv_serror_rate and should be ignored for analysis.
#(Correlation = 0.9993041091850098)
df.drop('dst_host_srv_serror_rate',axis = 1, inplace=True)

#This variable is highly correlated with rerror_rate and should be ignored for analysis.
#(Correlation = 0.9869947924956001)
df.drop('dst_host_serror_rate',axis = 1, inplace=True)

#This variable is highly correlated with srv_rerror_rate and should be ignored for analysis.
#(Correlation = 0.9821663427308375)
df.drop('dst_host_rerror_rate',axis = 1, inplace=True)

#This variable is highly correlated with rerror_rate and should be ignored for analysis.
#(Correlation = 0.9851995540751249)
df.drop('dst_host_srv_rerror_rate',axis = 1, inplace=True)

#This variable is highly correlated with dst_host_srv_count and should be ignored for analysis.
#(Correlation = 0.9865705438845669)
df.drop('dst_host_same_srv_rate',axis = 1, inplace=True)
```

Figure 3.4: Dropping uncorrelated features

From figure 3.4, we reduced the number of features to 33 from the original 43. The 33 features were the most correlated and are finally used for the implementation. Figure 3.5

```
['duration', 'protocol_type', 'service', 'flag', 'src_bytes',
 'dst_bytes', 'land', 'wrong_fragment', 'urgent', 'hot',
 'num_failed_logins', 'logged_in', 'num_compromised', 'root_shell',
 'su_attempted', 'num_file_creations', 'num_shells', 'num_access_files',
 'is_guest_login', 'count', 'srv_count', 'serror_rate', 'rerror_rate',
 'same_srv_rate', 'diff_srv_rate', 'srv_diff_host_rate',
 'dst_host_count', 'dst_host_srv_count', 'dst_host_diff_srv_rate',
 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'target',
 'Attack Type'],
```

Figure 3.5: Selected features

V. METHODOLOGY AND IMPLEMENTATION

The datasets shown in figures 3.1 and 3.2 were used in the development of the chosen machine learning algorithms in order to meet the study's objectives. Figure 3.6 displays the stages involved in the complete procedure.

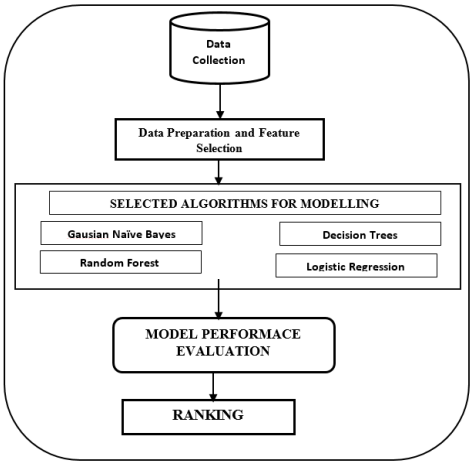

Figure 3.6: Overview of the system

The system's implementation utilized the Python programming language within the Jupyter Notebook framework. The necessary python classes for the selected machine learning algorithms (Naïve Bayes, Decision trees, random forest and logistic regression) were imported, and subsequent model modelling. Following experimentation, the evaluation outcomes were studied from which their performance rankings were carried out as outlined in the following subsections..

A. Result and Evaluation

The four machine learning models were evaluated against the most commonly used metrics for classification problems, these are; cross validation mean score, model accuracy, precision, recall and f1-score. The summary of the results is presented in table 1.

Table 1: Evaluation Metrics

Model	Accuracy	Cross Val	Precision		Recall		F1-Score	
			Anomaly	Normal	Anomaly	Normal	Anomaly	Normal
Gaussian Naïve Bayes	0.90	0.908	0.95	0.88	0.95	0.96	0.90	0.92
Decision Trees	0.98	0.99	1.00	0.99	1.00	0.99	0.99	0.98
Random Forest	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
Logistic Regression	0.94	0.95	0.96	0.95	0.94	0.97	0.96	0.96

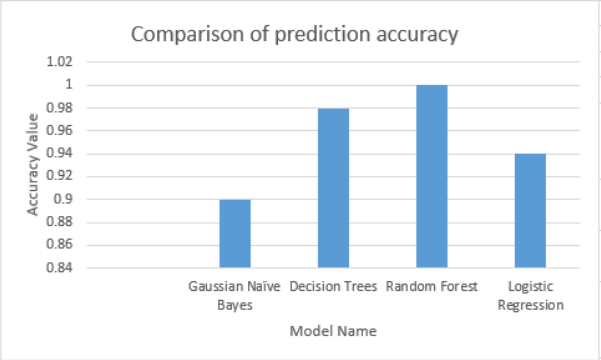

Figure 3.6: Comparison of the model's prediction accuracy

The classification models' performances were assessed by comparing various metrics detailed in Table 1. Analyzing the data presented in Table 1 alongside Figure 3.6 distinctly identifies the most efficient algorithm. Results indicate that the random forest algorithm exhibited the highest level of prediction accuracy, whereas the Gaussian Naïve Bayes algorithm demonstrated comparatively lower performance.

B. Comparison with Existing Works

This research stands out from existing literature by offering a unique perspective through a comparative analysis of multiple classification algorithms. What sets this study apart is its systematic evaluation of these algorithms using the same dataset, allowing for a direct comparison of their performances. To underscore the efficacy of our approach, we juxtapose the evaluation metrics from previous studies, providing a comprehensive comparative analysis presented in Table 2.

Table 2: Evaluation

Reference	Method Used	Accuracy	This work	
Ashiku & Dagli (2021).	Deep Learning	94%	Decision Trees Random Forest Logistic Regression	98% 100% 94%
Mebawondu *et al* (2020)	Artificial Neural Network	76.96%	Decision Trees Random Forest Logistic Regression	98% 100% 94%
Das *et al* (2010)	Support Vector Machine (Rough set theory)	93%	Decision Trees Random Forest Logistic Regression	98% 100% 94%

CONCLUSION

This research work primarily aimed to build machine learning models tailored for network intrusion detection, with a core focus on accurately classifying connections as either suspicious or normal. Alongside this primary objective, an equally significant goal was to conduct a comprehensive comparative analysis of the selected algorithms, culminating in their ranking based on performance metrics. The experimental findings underscored the efficacy of the random forest classifier as the optimal machine learning algorithm for network intrusion detection. Through experimentation and evaluation with the selected dataset, the random forest algorithm demonstrated higher capability in terms of prediction accuracy. Furthermore, the comparative study shed light on the strengths and limitations of each algorithm, providing valuable insights for future research and practical applications in network security.

REFERENCES

[1] Acheme, I. D., & Vincent, O. R. (2021). Machine-learning models for predicting survivability in COVID-19 patients. In Data Science for COVID-19 (pp. 317-336). Academic Press.

[2] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.

[3] Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. *Procedia Computer Science*, *185*, 239-247.

[4] Bakhsh S, Alghamdi S, Alsemmeari RA, Hassan SR. (2019) An adaptive intrusion detection and prevention system for Internet of Things. International Journal of Distributed Sensor Networks. 2019;15(11). doi:10.1177/1550147719888109.

[5] Das, V., Pathak, V., Sharma, S., Srikanth, M. V. V. N. S., Kumar, G., & Nadu, T. (2010). Network intrusion detection system based on machine learning algorithms.

[6] De'ath, G., & Fabricius, K. E. (2000). Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology*, *81*(11), 3178-3192.

[7] Gautam, R. K. S., & Doegar, E. A. (2018, January). An ensemble approach for intrusion detection system using machine learning algorithms. In *2018 8th International conference on cloud computing, data science & engineering (confluence)* (pp. 14-15). IEEE.

[8] Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, *5*(2), 1.

[9] Jamadar, R. A. (2018). Network intrusion detection system using machine learning. *Indian Journal of Science and Technology*, *7*(48), 1-6.

[10] Kumar Singh Gautam R. and Doegar, E. A. (2018) "An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms," 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018, pp. 14-15, doi: 10.1109/CONFLUENCE.2018.8442693.

[11] MAKINDE, A. S., & ACHEME, I. D. (2023). Climate-Driven Maize Yield Prediction: A Machine Learning Approach.

[12] Manavalan, B., Subramaniyam, S., Shin, T. H., Kim, M. O., & Lee, G. (2018). Machine-learning-based prediction of cell-penetrating peptides and their uptake efficiency with improved accuracy. *Journal of proteome research*, *17*(8), 2715-2726.

[13] Mebawondu, J. O., Alowolodu, O. D., Mebawondu, J. O., & Adetunmbi, A. O. (2020). Network intrusion detection system using supervised learning paradigm. *Scientific African*, *9*, e00497.

[14] Patgiri, R. Varshney, U. Akutota, T., and Kunde, (2018) "An Investigation on Intrusion Detection System Using Machine Learning," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1684-1691, doi: 10.1109/SSCI.2018.8628676.

[15] Tahri, R., Balouki, Y., Jarrar, A., & Lasbahani, A. (2022). Intrusion Detection System Using machine learning Algorithms. In *ITM Web of Conferences* (Vol. 46, p. 02003). EDP Sciences.

[16] Thomas Rincy N, Roopam Gupta, (2021) "Design and Development of an Efficient Network Intrusion Detection System Using Machine Learning Techniques", *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9974270, 35 pages, 2021. https://doi.org/10.1155/2021/9974270

[17] Varanasi V. and Razia S. (2022) "Network Intrusion Detection using Machine Learning, Deep Learning - A Review," *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1618-1624, doi: 10.1109/ICSSIT53264.2022.9716469.

[18] Yadav M. K. and Sharma K. P. (2021) “Intrusion Detection System using Machine Learning Algorithms: A Comparative Study,” 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), 2021, pp. 415-420, doi: 10.1109/ICSCCC51823.2021.9478086.