Research Article *Journal of Contemporary Education Theory & Artificial Intelligence*

A Comparative Analysis of Network Intrusion Detection Using Different Machine Learning Techniques

Authors:

Siddharth Konkimalla[1*], Gagan Kumar Patra[2], Chandrababu Kuraku[3], Janardhana Rao Sunkara[4], Sanjay Ramdas Bauskar[5], Chandrakanth Rao Madhavaram[6], Kiran Polimetla[7]

[1]Adobe Inc, Sr Network Development Engineer. Email: Siddharth.konkimalla@gmail.com
[2]Tata Consultancy Services, Senior Solution Architect. Email: gagankpatra@outlook.com
[3]Mitaja Corportaion, Senior Solution Architect. Email: chandrababu.kuraku@gmail.com
[4]CVS Pharmacy Inc, Sr. Oracle Database Administrator. Email: Janardhanasunkara9@gmail.com
[5]Pharmavite LLC, Sr. Database Administrator. Email: sanjaybauskar@gmail.com
[6]Microsoft, Support Escalation Engineer. Email: Chandrakanthmadhavaram@gmail.com
[7]Adobe Inc, Software Engineer. Email: Kiran.polimetla@gmail.com

[*]**Corresponding author:** Siddharth Konkimalla, Adobe Inc, Sr Network Development Engineer.

Citation: Siddharth K, Gagan Kumar P, Chandrababu K, Janardhana Rao S, Sanjay Ramdas B, et al. (2023) A Comparative Analysis of Network Intrusion Detection Using Different Machine Learning Techniques. J Contemp Edu Theo Artific Intel: JCETAI-102.

Received Date: 10 October, 2023; **Accepted Date:** 18 October, 2023; **Published Date:** 23 October, 2023

Abstract

The security of modern communication networks cannot be adequately assured without intrusion detection systems (IDS). Pattern recognition, signature analysis, and rule violation detection were the primary goals of these systems. Recent advances in ML and DL approaches have shown promise as prospective replacements in a field of NID. Typical and anomalous patterns may be distinguished using these techniques. This paper uses the NSL-KDD benchmark data set to assess NIDS using many ML algorithms, like SVM, DT, LR, and RF classification. We evaluate the precision, recall, accuracy, and FPR of several ML techniques, such as SVM accuracy is 98.97%, Random Forests, and Decision Trees. The results demonstrate that, in comparison to conventional techniques, machine learning approaches greatly increase detection rates while reducing false alarms. And in this, a RF achieve a high accuracy which is 99.83%. The results of this investigation demonstrate that not only is it feasible to obtain a high detection rate of assaults, but also accurate prediction. It is clear from these findings that ML has great promise for developing highly efficient NIDS systems.

Keywords: *Network Intrusion Detection (NID), Machine Learning, NSL-KDD, Support Vector Machines, Random Forests, and Decision Trees.*

Introduction

In recent years, networks have become more important to the contemporary way of living. As a result, cybersecurity has emerged as a lucrative field of study that has inspired fresh ideas in data innovation [1][2]. Protecting private information passing via networks is increasingly essential to contemporary society, since networks, and security in particular, are among the most critical concerns in the area of information security. Software such as firewalls, IDS, and antivirus programs are the core components of cybersecurity techniques. Computer engineers still face several difficulties, however, since hackers and other hackers may successfully try to breach computer systems or networks. Consequently, there is a growing demand in this industry to develop more potent IDS [3].

The development of the internet in this age has positively affected several endeavours. Emerging massive data and information sets are likewise affected by this enhancement [4]. As the internet evolves, several challenges must be addressed to make it a more reliable, stable, and secure system. Firewalls, certain dynamic processes, software, etc., are only a few of the many options available for making systems more secure. IDS are among the most effective dynamic mechanisms for identifying and stopping certain types of network intrusions [4]. The primary function of an IDS is to keep an eye on network processes and analyse them for any signs of unusual activity or divergence from the norm [5][6]. Programs are made to scan network data for indications of harmful activity or infringements of policies. NIDS, HIDS, PIDS, APIDS, and HIDS consist of five distinct types of IDS. The two primary detection approaches are signature-based detection and anomaly detection, which is also called misuse detection [7][8].

IDS are classified into five types: APIDS, NIDS, PIDS, HIDS, and HIDS. Misuse detection is otherwise called signature-based detection where machine learning Strengthen the architectures of these systems and enables two pivotal detections Figure 1. Misuse detection employs machine learning techniques to match the present traffic with previously learned attack patterns, while anomaly detection applies machine learning to discover new traffic normality or the lack of it [9][10]. Through integration of the ML models, IDS will be able to analyse new threats in order to enhance the detection rate while also enhancing the response time of the system against known and unknown threats [11].

The focus of this study is to identify and analyse multiple approaches to the detection of network intrusions and improve, in general, the security of network systems in relation to the constantly emerging and more complex cyber threats. The study aims to contribute to the following goals by analysing distinct techniques like DT, SVM, and NN, while attempting to discover the best approach to recognise known and unknown intrusions. Furthermore, this study also seeks to evaluate the effectiveness of these techniques when implemented on benchmark datasets including, but not limited to NSL-KDD, taking into account factors like accuracy and time complexity of the process with an eventual view of deploying it in real life. More effective IDSs for safeguarding network infrastructure are the ultimate goal of this project. The primary findings of the study are as follows:

- A research that compares several ML approaches, including RF and SVM, for NID.
- Utilization of the NSL-KDD dataset to address previous dataset limitations and enhance testing robustness.
- Demonstration that Random Forest significantly outperforms other models in F1-score, recall, accuracy, and precision.
- Effective use of SMOTE to address class imbalance and enhance model performance.
- Provides insights and recommendations for advancing machine learning approaches in network security.

A. Structure of Paper

This study is organised as follows: In Section II, the prior research is summarised. The study methodology is detailed in Section III, which also includes the classification models utilised for analysis. The experimental data are detailed and analysed in Section IV, with an emphasis on the performance indicators for each model. Findings and recommendations for further study are presented in Section V.

Literature Review

This section examines a range of literature centred on network intrusion detection, emphasising significant studies that investigate various methodologies for NIDS. The most relevant research publications on this topic are summarised in Table 1.

In this paper, Abraham and Bindu (2021) this research aim to investigate various DL and ML approaches to intrusion detection by analysing existing research and providing context on these algorithms as they pertain to IDS. A performance comparison of several ML classification techniques using the DARPA dataset is also included in the paper. An IDS's performance is based on how accurate it is. Raising detection rates while decreasing false alarms requires improved intrusion detection accuracy [12].

In this paper, Disha and Waheed, (2021) make employ of ML methods to construct IDS, since ML models effectively provide improved accuracy in detecting anomalies. However, in order to test the ML models that relied on binary classification, they employed the UNSW-NB 15 dataset, which is available offline. The DT, RF, GBT, and MLP were trained and tested in order to undertake performance analysis. They eliminated the characteristics that were unrelated to response employing a Chi-Square test. A result showed that DT was a most accurate classifier, with the lowest FPR. Feature deletion increased the overall performance of all models except RF. Our suggested strategy outperformed other current ML algorithms in terms of accuracy, according to experimental study [13].

In this paper, Halimaa and Sundarakantham (2019) different kinds of IDS have been developed to safeguard networks using a variety of ML and statistical methodologies. This issue is addressed in the suggested method. ML methods like SVM and NB are used. Using the NSL-KDD knowledge discovery dataset, an IDS may be evaluated [14].

In this paper, Chabathula, Jaidhar and Ajay Kumara, (2015) PCA is used to convert datasets with greater dimensions into datasets with fewer dimensions. SVM, KNN, J48 Tree algorithm, RF classification algorithm, Adaboost algorithm, Nearest Neighbours generalised Exemplars algorithm, NB probabilistic classifier, and Voting Features Interval classification algorithm are test methods used for the reduced dimension dataset. KDD 99 is the data set used throughout the whole experimen t[15].

In this paper, Aljohani and Bushnag, (2021) The KDD99 dataset is used to test the suggested method. When it comes to anomaly-based detection, the KDD99 is the gold standard. This method effectively and quickly detects assaults. When compared to all of the SVM kernel models, Neural Network demonstrated superior classification accuracy. Prevention of LAN security threats is the goal of the proposed approach, which employs SVM and NN intrusion detection models [16].

Table 1: Presents comparative table on network Intrusion detection using machine learning.

References	Methodology	Dataset	Performance	Limitations & Future Work
[12]	In-depth review of DL and ML methods for intrusion detection; comparison of ML classification methods	DARPA dataset	Comparison of various ML classification methods; performance based on accuracy	Need to improve intrusion detection accuracy to decrease false alarms and increase detection rates
[13]	Machine learning techniques (DT, RF, GBT, MLP); feature elimination with Chi-Square test	UNSW-NB 15 dataset	DT showed maximum accuracy and lowest FPR; overall performance improved feature elimination.	Limitations in other models like RF; Future work should explore other feature selection techniques and advanced ML models
[14]	Machine learning techniques (SVM, Naïve Bayes) for classification problems	NSL-KDD dataset	SVM and Naïve Bayes were applied, with emphasis on accuracy	Future work may involve exploring other datasets and advanced classification methods to further enhance detection accuracy
[15]	Principal Component Analysis (PCA) for dimensionality reduction; various classification	KDD99 dataset	TREE classification algorithms showed superior detection	Future work could include testing other datasets and improving system resource utilisation.

	algorithms (SVM, KNN, J48, RF, Adaboost, etc.)		accuracy, computational efficiency, and low false alarms	
[16]	Comparison of SVM and Neural Network models for anomaly-based detection	KDD99 dataset	Neural Networks outperformed SVM models, especially in classification accuracy.	Future work may involve optimising Neural Network models and exploring hybrid models for better efficiency.

B. *Research gaps*

Despite significant progress, several research gaps remain in the area of IDS that might benefit from DL and ML techniques. One notable issue is that most research focuses on particular datasets, like KDD99, NSL-KDD, or UNSW-NB 15, which cannot adequately depict a diversity of contemporary network traffic. As a result, models' generalizability across multiple datasets is limited. Furthermore, even with the great accuracy achieved by many techniques, false-positive rates remain a difficulty, resulting in unreliable detection in practical circumstances. Additionally, despite the promising findings of neural networks and other advanced models, their implementation in resource-constrained contexts is limited due to their processing cost. Lastly, research into creating scalable, effective, and reliable IDS solutions is still needed, as the integration of hybrid models and real-time adaptive mechanisms to dynamically increase detection performance is still in its early stages.

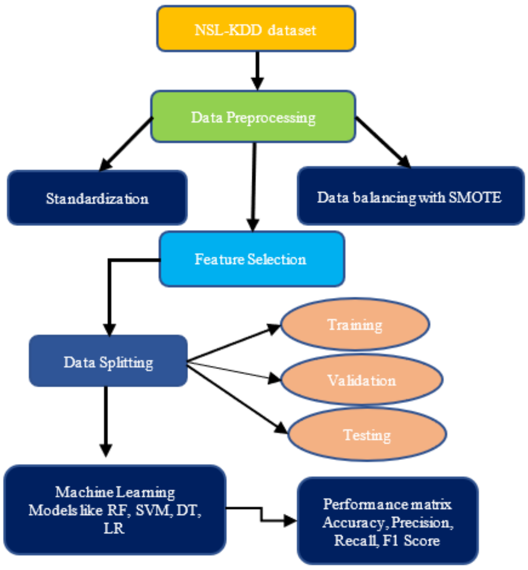
NSL-KDD dataset
Data Preprocessing
Standardization
Data balancing with SMOTE
Feature Selection
Data Splitting
Training
Validation
Testing
Machine Learning Models like RF, SVM, DT, LR
Performance matrix Accuracy, Precision, Recall, F1 Score

Figure 1: Data Flow Diagram of methodology for network intrusion detection system.

Methods and Materials

This study aims to find the best ML methods for network intrusion detection by comparing and contrasting several methods. The end objective is to find the best method for improving network system security by differentiating between safe and dangerous actions most rapidly and correctly. This research study involves several key steps, beginning with data collection using the NSL-KDD dataset, which addresses issues from the KDD 99 benchmark and contains 41 features. Data preprocessing is performed, which includes standardisation to normalise feature values and ensure consistent scaling, followed by feature selection using correlation analysis to eliminate redundant features. The SMOTE is used to generate synthetic data points after rectifying an issue of class imbalance. To prevent overfitting, the dataset is distributed as follows: training, validation, and testing sets, with a ratio of 70:15:15. There are a number of ML algorithms used for training models, such as RF, SVM, DT, and LR. Multiple decision trees are built using randomised data sets in the RF model; SVM is used for fast classification in DT; entropy is used to define classification rules in RF; and the likelihood of binary outcomes is modelled using LR. Each model's performance is evaluated employing metrics like as Recall, F1-score, precision, and accuracy. Figure 1 is a flow diagram depicting the network intrusion detection approach.

The steps in the data flow diagram are outlined below, providing a detailed explanation of each stage involved in the system's data processing.

A. *Data Collection*

The NSL-KDD dataset, which overcomes problems with the KDD 99 benchmark, was used for data collection for this work. The dataset consists of connection records with 41 features, including 34 numeric and 7 symbolic or discrete features. The NSL-KDD training set has 22 different attack kinds, whereas the testing set contains an additional 17 attack types that were not included in the training set.

B. *Data Preprocessing*

Data preparation for analysis or modelling is called preprocessing. Data preparation is a process of improving the quality and analytical applicability of data by cleaning, converting, and organising it. Common tasks include filling in missing values, eliminating duplicates, standardising data, and encoding categorical variables. The goal is to make data analysis and ML models more accurate and efficient.

C. *Standardization:*

A crucial approach to feature scaling is standardisation, which is often called z-score normalisation. The process entails dividing the value of each characteristic by its standard deviation after removing the mean. In cases when the input data has a wide range of feature values, this method shines [17]. After being standardised, all features are on the same scale, with a mean (μ) of 0 and a standard deviation (σ) of 1.

$$x_{new}=\frac{x-\mu}{\sigma} \qquad (1)$$

The accuracy of our prediction models is much improved by this procedure. Normalisation of the Z-score mathematically represented in Equation (1).

D. *Feature Selection*

Feature selection is a method for improving and streamlining subgroups by removing irrelevant or superfluous properties and focussing on the most important ones [18]. Correlation is a well-liked and effective method for finding the most related features in any dataset; it establishes the degree of association between features on the premise that they are conditionally independent with respect to the class. Characteristics that are highly predictive of the class and not predictive of each other make up a strong feature subset.

E. Data balancing with SMOTE

Class imbalance in datasets may be addressed using the SMOTE approach, which creates synthetic samples for the minority class. It creates new instances by interpolating between existing samples, helping improve model performance in classification tasks where imbalanced data could lead to biased results[19].

F. Data splitting

In data division, data splitting is a crucial step. The dataset used in this study has been separated into three sub-sections: 70% served for training, 15% had been used for validation, and 15% was used for testing. This approach guarantees that the model will learn from one subset and be evaluated on another, hence reducing overfitting.

G. Classification Models

This section outlines the ML models employed for classification using a NSL-KDD dataset and evaluates their performance to determine their effectiveness.

1) Random forest (RF)

The RF algorithm uses the ensemble learning approach for classification and regression. This method is designed for supervised learning. It uses a combination of n regression trees to provide more accurate predictions than a single tree could on its own. When training, RF constructs a forest of decision trees, which it then uses to make a final prediction by combining their predictions. Data scientists may use RF to lower the variance of algorithms, especially DT, that have a large variation by using random sampling with replacement, or bagging in ML terminology[20]. Bagging takes a training set of features X and outputs Y, then iteratively fits the trees to random samples from a training set β times (b=1, 2,….,β).

A replacement set of cases is obtained for each tree by randomly sampling them from a training set. Every set of occurrences represents a unique tree via a random vector Øk. The decision trees built from these sequences will also vary significantly as they will not be identical. It is proposed that Equation (2) may be used to describe a K-th tree's forecast for an input X:

$$h_k(\mathrm{X}) = \mathrm{h}(\mathrm{X}, \emptyset_k),), \forall k \in \{1,2,\ldots.,K\} \qquad (2)$$

where K is a total number of trees. During a tree's branching process, every node picks characteristics at random to minimise feature correlations.

2) *Support vector machine (SVM)*

A popular ML technique for regression and classification problems is the SVM. SVM was used in cheminformatics and bioinformatics, among other fields. Using training data, the SVM classifier creates a model for the classification. The categorisation of an unidentified sample is a subsequent step [21]. The core principle of SVMs is the use of hyperplanes to establish hierarchies. When the data can be divided linearly, SVM has shown impressive accuracy. Non-linear separation of separable data is not possible using SVM output.

3) *Decision tree (DT)*

The DT algorithm is a well-recognised technique for classification. A decision tree graph resembles a tree. Based on the criteria that are implemented from the tree's root to its leaf, it classifies objects. The test nodes are located within the network, the branches represent the test results, and the leaf nodes determine the categorisation. A data set is selected based on its purity level. The quantification of this impurity is done using entropy. A high entropy level indicates a high level of impurities [22].

4) Logistic Regression (LR)

LR is a classification approach that assumes that the result is influenced by several independent factors. To determine the likelihood of an event occurring, LR applies a probability function; it is a kind of binary classification [23]. It computes the probability using the formula below. Among the benefits are its quick classification speed and ease of extension to multi-class problems. The primary drawback is that LR cannot be used to handle nonlinear problems[24].

A. Performance matrix

A number of measures were used to evaluate the model's performance, including recall, accuracy, precision, F1-score, and ROC curve. These measurements make it possible to assess every class separately. Below are the formulae needed to calculate these performance metrics.

1) Accuracy:

The percentage of all forecasts that were accurate is known as the accuracy (AC). It may be found in Equation (3):

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

2) Precision:

The ratio of real positives to the total of both real and false positives is one way to describe the precision. Equation (4) provides the following:

$$Precision = \frac{TP}{TP+FP} \qquad (4)$$

3) Recall:

The word "recall" describes the proportion of correctly classified positive cases as a fraction of all positive examples. Equation (5) provides the mathematical expression for it:

$$Recall = \frac{TP}{TP+FN} \qquad (5)$$

4) F1-score:

A harmonic mean of recall and precision in a classification task is measured by the F1-score. This is given by Equation (6):

$$\text{F1- Score} = 2.\, \frac{(Precision\,.\ Recall)}{Precision + Recall)} \qquad (6)$$

Each of the four cells that comprise the output matrix represents a different outcome: TP, TN, FP, or FN. A positive relationship between the actual and projected values is shown by TP; TN occurs when the model predicts negative values while the data shows positive ones; FP indicates that the model predicts positive values whereas in fact the results are negative; and finally, a negative value for both the anticipated and actual values is denoted by FN.

5) ROC-AUC

The most crucial metric for evaluating the model is an area under a ROC curve, which is often abbreviated as AUC. Each time, a TPR and FPR were computed as the horizontal and vertical axes, respectively, based on the sorted prediction results of the model, which indicated that the samples were forecasted as positive instances in a certain sequence.

Result Analysis and Discussion

Results from testing ML models on the NSL-KDD dataset for detecting network intrusions are shown here. In addition, compare and contrast the different NIDS ML models using f1-score, recall, precision, and accuracy metrics.

Table 2: Results of Random Forest model for NIDS.

Performance Measures	Random Forest
Accuracy	99.83
Precision	99.93
Recall	99.71
F1-Score	99.82

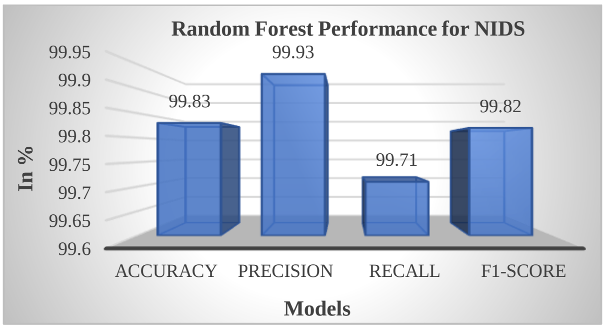

Figure 2: Results of Random Forest for NIDS.

Figure 2 illustrates a performance of the RF model, which achieved a highest result across all classification metrics. The graph displays key metrics like Recall, accuracy, precision, and F1-score. On the x-axis, a various performance metrics are presented, while a y-axis shows a corresponding metric values. The model demonstrated strong performance with an accuracy 99.83, precision 99.93, recall 99.71 and F1-score 99.82. Overall, a Random Forest model excelled in all evaluation metrics, highlighting its effectiveness in accurately classifying the data.

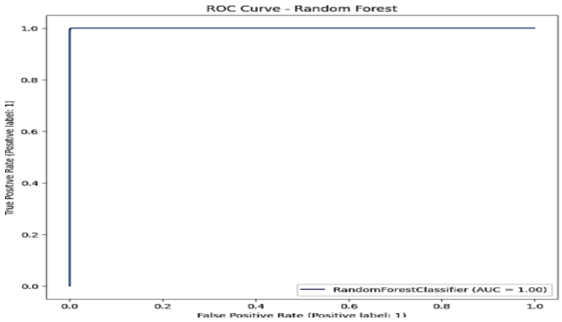

Figure 3: ROC curve of random forest model.

Figure 3 represents a ROC curve for an RF model. Plotting the TPR versus the FPR is what the curve does. The model performs quite well in differentiating among positive and negative classes, as evidenced by its AUC of 1.00, respectively.

A. Comparative Analysis

This Table displays the results of several ML algorithms that were run on the NSL-KDD dataset in order to analyse NID. The following is an examination and explanation of several ML models' performance metrics, including Accuracy, Precision, Recall, and F1-score:

Table 3: Comparison between various machine learning models for the analysis of network intrusion detection.

Model	Accuracy	Precision	Recall	F1-Score
SVM[25]	98.97	99.9	99.2	99.2
LR[26]	81.51	0.851	0.815	0.832
DT[27]	0.82	0.82	0.82	0.80
RF	99.83	99.83	99.83	99.83

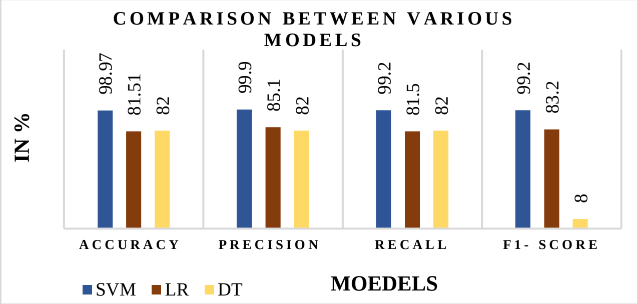

Figure 4: Comparison between various models.

Table 3 and Figure 4 also contains the outcomes of the comparison of a comparative models for the analysis of NID classification models. A present study focuses on a comparison of a machine learning models, SVM, LR, DT, RF, and identifies that the RF model is effective for intrusion detection with high precision, recall, accuracy, an F1-score of 99.83%. Next is Support Vector Machine which performs comparably to k-NN with overall accuracy of 98.97%, and precision of 99.9% which shows good classification capacity. As can be seen, the performance of the LR and DT models is much lower – LR has the accuracy of 81.51 %, while DT has the lowest result of 82%. Here LR shows only moderate levels of accuracy and recall compared to other methods such as SVM or RF. Nevertheless, the assessment of all of the metrics taken into consideration proves that the RF model is the most efficient and reliable one among all the other types of the model and, consequently, followed by the SVM model.

Conclusion and Future Scope

Security is seen as a primary problem of the network due to the increasing use of network services. Numerous networked computers are vital to the operation of businesses and other applications that rely on the network to provide services. As a result, this study evaluated the usefulness of the relevant framework algorithms when applied to the NSL-KDD dataset and suggested a NIDS based on the use of ML techniques. It is evident from the comparative study that only highly skilled IDS are capable of preserving the network's integrity. The study also reveals that the suggested method minimised false positive rates and obtained excellent detection accuracy, with DT, RF, and SVM models performing the best overall with an accuracy of 98.97%. An overview of future works is presented in the report where more detailed analysis of these algorithms has to be produced in the case of the multiclass classification and real-time applications.

Further research should be directed towards the studies of mixed and combined methods, the feature selection techniques, and solutions regarding the large-scale problem to combat contemporary network threats. Besides, the use of IDS with higher efficiency as well as ability to develop concrete actions against new types of threats will imply the integration of deep learning approaches and their application in various datasets. In conclusion, this study reveals that there is a continuous demand for new ideas into intrusion detection, which lays the foundation to future developments in Network Security.

References

1. K. A. Taher, B. Mohammed Yasin Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," in *1st International Conference on Robotics, Electrical and Signal Processing Techniques, ICREST 2019*, 2019. doi: 10.1109/ICREST.2019.8644161.
2. V. K. Yarlagadda and R. Pydipalli, "Secure Programming with SAS: Mitigating Risks and Protecting Data Integrity," *Eng. Int.*, vol. 6, no. 2, pp. 211–222, 2018.
3. B. A. Tama and K. H. Rhee, "An extensive empirical evaluation of classifier ensembles for intrusion detection task," *Comput. Syst. Sci. Eng.*, vol. 32, no. 2, pp. 149–158, 2017.
4. S. G. Priya Pathak, Akansha Shrivastava, "A survey on various security issues in delay tolerant networks," *J Adv Shell Program.*, vol. 2, no. 2, pp. 12–18, 2015.
5. V. Pai, Devidas, and N. D. Adesh, "Comparative analysis of Machine Learning algorithms for Intrusion Detection," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1013, no. 1, p. 012038, Jan. 2021, doi: 10.1088/1757-899X/1013/1/012038.
6. V. V. Kumar, S. R. Yadav, F. W. Liou, and S. N. Balakrishnan, "A digital interface for the part designers and the fixture designers for a reconfigurable assembly system," *Math. Probl. Eng.*, 2013, doi: 10.1155/2013/943702.
7. M. Rai and H. L. Mandoria, "Network Intrusion Detection: A comparative study using state-of-the-art machine learning methods," in *IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, ICICT 2019*, 2019. doi: 10.1109/ICICT46931.2019.8977679.
8. M. R. Kishore Mullangi, Vamsi Krishna Yarlagadda, Niravkumar Dhameliya, "Integrating AI and Reciprocal Symmetry in Financial Management: A Pathway to Enhanced Decision-Making," *Int. J. Reciprocal Symmetry Theor. Phys.*, vol. 5, no. 1, pp. 42–52, 2018.
9. J. Thomas and V. Vedi, "Enhancing Supply Chain Resilience Through Cloud-Based SCM and Advanced Machine Learning: A Case Study of Logistics," *J. Emerg. Technol. Innov. Res.*, vol. 8, no. 9, 2021.
10. V. Bhatia, S. Choudhary, and K. R. Ramkumar, "A Comparative Study on Various Intrusion Detection Techniques Using Machine Learning and Neural Network," in *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 2020. doi: 10.1109/ICRITO48877.2020.9198008.
11. S. Anwar *et al.*, "From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions," *Algorithms*. 2017. doi: 10.3390/a10020039.
12. J. A. Abraham and V. R. Bindu, "Intrusion Detection and Prevention in Networks Using Machine Learning and Deep Learning Approaches: A Review," in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2021, pp. 1–4. doi: 10.1109/ICAECA52838.2021.9675595.
13. R. A. Disha and S. Waheed, "A Comparative study of machine learning models for Network Intrusion Detection System using UNSW-NB 15 dataset," in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, 2021, pp. 1–5. doi: 10.1109/ICECIT54077.2021.9641471.
14. A. A. Halimaa and K. Sundarakantham, "Machine learning based intrusion detection system," in *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, 2019. doi: 10.1109/ICOEI.2019.8862784.
15. K. J. Chabathula, C. D. Jaidhar, and M. A. Ajay Kumara, "Comparative study of Principal Component Analysis based Intrusion Detection approach using machine learning algorithms," in *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015, pp. 1–6. doi: 10.1109/ICSCN.2015.7219853.
16. A. Aljohani and A. Bushnag, "An Intrusion Detection System Model in a Local Area Network using Different Machine Learning Classifiers," in *2021 11th International Conference on Advanced Computer Information Technologies (ACIT)*, 2021, pp. 483–488. doi: 10.1109/ACIT52158.2021.9548421.
17. K. Patel, "Quality Assurance In The Age Of Data Analytics: Innovations And Challenges," *Int. J. Creat. Res. Thoughts*, vol. 9, no. 12, pp. f573–f578, 2021.
18. N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3167918.3167951.
19. J. H. Seo and Y. H. Kim, "Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection," *Comput. Intell. Neurosci.*, 2018, doi: 10.1155/2018/9704672.
20. N. Farnaaz and M. A. Jabbar, "Random Forest Modeling for Network Intrusion Detection System," in *Procedia Computer Science*, 2016. doi: 10.1016/j.procs.2016.06.047.
21. D. M. Abdullah and A. M. Abdulazeez, "Machine Learning Applications based on SVM Classification: A Review," *Qubahan Acad. J.*, 2021, doi: 10.48161/qaj.v1n2a50.
22. A. Pathak and S. Pathak, "Study on Decision Tree and KNN Algorithm for Intrusion Detection System," *Int. J. Eng. Res.*, vol. 9, no. 5, 2020.
23. V. Kumar and F. T. S. Chan, "A superiority search and optimisation algorithm to solve RFID and an environmental factor embedded closed loop logistics model," *Int. J. Prod. Res.*, vol. 49, no. 16, 2011, doi: 10.1080/00207543.2010.503201.
24. E. Besharati, M. Naderan, and E. Namjoo, "LR-HIDS: logistic regression host-based intrusion detection system for cloud environments," *J. Ambient Intell. Humaniz. Comput.*, 2019, doi: 10.1007/s12652-018-1093-8.

25. V. Pai, Devidas, and N. D. Adesh, "Comparative analysis of Machine Learning algorithms for Intrusion Detection," in *IOP Conference Series: Materials Science and Engineering*, 2021. doi: 10.1088/1757-899X/1013/1/012038.
26. A. M. Mahfouz, D. Venugopal, and S. G. Shiva, "Comparative Analysis of ML Classifiers for Network Intrusion Detection," no. Ml, pp. 1–13, 2019.
27. R. Ahsan, W. Shi, and J.-P. Corriveau, "Network intrusion detection using machine learning approaches: Addressing data imbalance," *IET Cyber-Physical Syst. Theory Appl.*, vol. 7, 2021, doi: 10.1049/cps2.12013.