Comparative Analysis of Machine Learning Algorithms for Intrusion Detection System

[1]Vikash, [2] Ms. Suruchi

[1]M.Tech. Scholar, [2]Assistant Professor

Department of CSE, BRCM CET, Bahal, Bhiwani, Haryana (India)

[1]vikashpilania60@gmail.com

[2]suruchi@brcm.edu.in

ABSTRACT

IDS (Intrusion Detection Systems) are critical for protecting networks of computers from hostile activities. The necessity for reliable intrusion detection solutions has increased as cyber threats become more sophisticated. Because of their ability to learn patterns from big datasets, machine learning algorithms appear to be potential methods for improving IDS detection capabilities. Several machine-learning methods for intrusion detection, including supervised, unsupervised, and semi-supervised strategies, are thoroughly examined in this work. The study compares the performance of algorithms including Decision Trees, Random Forests, Support Vector Machines, K-Nearest Neighbours and Naive Bayes. Algorithm efficacy is evaluated by assessment criteria like recall, precision, accuracy, & F1-score, etc. Furthermore, the study investigates the strengths, limits, and application of various algorithms in diverse network traffic and attack scenarios. The findings of this investigation provide essential help in determining the best machine-learning technique for developing resilient and efficient Intrusion Detection Systems adapted to varied network landscapes.

***Keywords:** IDS (Intrusion Detection System), Machine Learning, Supervised Machine Learning Algorithms.*

INTRODUCTION

In today's digitally interconnected world, computer network security is critical for protecting sensitive data and maintaining operational continuity. The changing cyber threat landscape creates challenges, pushing firms to build strong defensive measures to prevent, compromised data breaches, unauthorized access & detrimental actions. IDS are critical because they actively monitor & Real-time network traffic analysis allowing them to recognize & address breaches in security. IDS are essential parts of the infrastructure for network security that identify and handle unauthorized access, abuse, and questionable activities. [1]. IDS can detect anomalies, patterns, or signatures that point to security concerns, allowing for prompt intervention and mitigation measures.

Deploying an IDS has various benefits, including:

1. Early threat detection: IDS can lessen the impact of malicious activity by swiftly identifying and responding to security concerns.

2. Improved security posture: Network traffic and system activity are continuously monitored by IDS, which assists businesses in identifying and resolving security risks early on.

3. Regulatory compliance: Many industry regulations and standards require intrusion detection systems as part of a

comprehensive cybersecurity strategy.

4. Incident response support: When IDS identify suspicious behavior, they provide alerts and notifications, enabling speedy investigation & reaction to security events by security teams.

IDS come in 2 categories: signature-based & anomaly-based.

1. Signature-based IDS: These systems, also referred to as knowledge-based intrusion detection systems, match observed system activity or network traffic to a database of pre-identified attack patterns or fingerprints. The IDS sends out a warning or initiates prearranged action to neutralize the threat when it discovers a match between the observed activity and a recognized attack signature. While excellent at identifying known threats, signature-based IDS may have trouble spotting new or undiscovered attacks.[1]

2. Anomaly-based IDS: Another name for it is behavior-based IDS, which is concerned with detecting anomalies in the course of normal network or system activity. By establishing a baseline of expected network activity, anomaly-based intrusion detection systems may recognize any behavior that differs from the norm as potentially suspicious. This method is quite effective at detecting previously unknown or zero-day threats, although it may occasionally create false positives if the baseline is poorly calibrated.[2]

Overview of the ML Algorithms For IDS

ML algorithms fall into a number of groups comprising:

1. Supervised Learning: Algorithm training is done through supervised learning using labeled datasets that comprise instances of both regular and invasive network traffic. Examples comprise 'Decision Trees, Random Forests, SVMs & Naive Bayes.' [1]
2. Unsupervised Learning: Conversely, unsupervised learning seeks to identify patterns or irregularities in the data without prior knowledge of intrusion occurrences and does not require labeled data. Clustering algorithms (like K-means & DBSCAN) are some of the techniques used.[3]
3. Semi-Supervised Learning: Labeled as well as unlabeled data are utilized to train models. This is helpful when labeled data is expensive or difficult to get. [4]
4. Reinforcement Learning: RL algorithms learn to make sequential decisions in response to environmental feedback.[5]
5. Hybrid Approaches: Hybrid approaches combine different machine learning techniques to maximize their strengths while mitigating the drawbacks of each method. This comprises ensemble methods (e.g., combining Decision Trees and SVM).[6]

Machine Learning Classifiers:

Decision Tree: Decision trees help categorize network traffic according to criteria inferred from historical data as either normal or invasive. [7]

Random Forest: Random Forest combines many decision trees to improve categorization accuracy. [8]

Support Vector Machines (SVM): These are adept at categorizing data, making them suitable for discriminating between regular and intrusive network traffic. [9]

K-Nearest Neighbors (KNN): It is a simple technique that allocates data points to the most prevalent class among their immediate neighbors. [10]

Naive Bayes: Naive Bayes classifiers are built on the Bayes theorem and depend on feature independence. They are capable of handling high-dimensional data and are computationally efficient.

Ensemble Methods: Techniques like AdaBoost and Gradient Boosting combine numerous weaker classifiers to create a powerful intrusion detection system.[11]

Data Reduction Methods:

ML & data mining encounter hurdles in intrusion detection due to huge and complicated datasets that necessitate significant computer resources for real-time deployment. The number of characteristics in network data challenges classification, as both number and quality affect accuracy and generalization. Feature extraction strategies have been shown to improve accuracy and processing efficiency when addressing these difficulties.[12]

1. Feature selection: These strategies attempt to improve performance by finding critical features, lowering computing time, and increasing accuracy, especially in Intrusion Detection Systems (IDS). Popular methods include PCA, IG, and GA, with two primary approaches: wrapper and filter methods.

Wrapper approaches: It uses classifiers to evaluate features, which may provide issues because of their high dimensionality.

Filter approaches: on the other hand, it uses separate estimation techniques such as distance and correlation measures, which provide robustness against overfitting.

2. Feature extraction: It uses approaches such as self-organizing maps and principal component analysis to minimize dataset dimensionality while retaining attack detection accuracy and speeding up discovery time.

3. Clustering: Clustering divides data samples into groupings based on their similarity in specific features.

Datasets Used:

The efficiency of machine learning in detecting anomaly threats is dependent on datasets. However, researchers continue to use outdated datasets like KDDCup99 & NSL-KDD, been criticized for their lack of relevance to modern network infrastructure. These datasets, created in 1999, fail to represent technological improvements like cloud computing and the Internet of Things.

1. KDDCup99: There are forty-one attribute-described connection objects in the KDDCup99 dataset, which was utilized in the 3rd International Knowledge Discovery & Data Mining Tools Competition. Each case is classified as either typical or suggestive of a specific kind of attack, and these can be further divided into 4 categories: probe, dos, U2R, or R2L.

2. NSL-KDD: An enhanced version of the KDDCup99 dataset is the 2009-created NSL-KDD dataset, with the goal of refining its structure by removing superfluous entries, correcting uneven instance counts, and reducing the range of attack classifications.

3. ISCX 2012: The 'University of New Brunswick's Information Security Centre of Excellence' (ISCX) produced the ISCX 2012 dataset in 2012. It provides binary classification without a specific attack type classification. Unfortunately, this dataset is currently unavailable and later renamed as CICIDS2017.

4. UNSW-NB15: IXIA PerfectStorm was used to simulate nine distinct types of attacks on the UNSW-NB15 dataset, which was generated by the Australia Centre for Cyber Security (ACCS). Fusers, analysis, backdoors, denial-of-service, exploits, generic, shellcode, reconnaissance, & worms were among the assaults that were used. with 47 attributes.[13]

Performance Metrics:

Performance metrics assess a model's performance on a particular dataset as well as its capacity to generalize to new data, both of which are critical for machine learning models.

True Positive (TP): When both the expected & original outputs were true, it is said to have occurred.

True Negative (TN): It is defined as the situation in which both the predicted and original outputs were false.

False Positive (FP): It is described as such when the predicted output is true but the actual output is false.

True Negative (TN): It is described as the situation where the actual output is true while the intended outcome is false.

Accuracy: The definition of it is calculated by dividing the total number of input samples by the number of accurate predictions.

$$\text{Accuracy} = \frac{TN + TP}{TP + TN + FP + FN}$$

Precision: By dividing the total number of accurate forecasts by the total number of positive forecasts, it is computed.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: It is defined as no. of true prediction divided by total positive samples.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F-Score: It is the precision and recall harmonic means.

$$\text{F-Score} = \frac{2(R*P)}{R + P}$$

$$\text{False Positive Rate} = \frac{FN}{TP + FN}$$

$$\text{False Negative Rate} = \frac{FP}{TP + FP}$$

Confusion Metrics: The table illustrates the performance of a classification algorithm by displaying the no. of false negatives, false positives, and real positives.

Table 1: Confusion Metrics

	Predicted Attack	Predicted Normal
Actual Attack	TP	FN
Actual Normal	FP	TN

LITERATURE REVIEW AND COMPARISON OF RELATED WORK :

Author [14] demonstrates that employing a single classifier for all types of attacks is not advisable, as different classifiers yield varying classification rates. The dataset used by this paper is KDD-NSL and used the following algorithms such as BF Tree, NB Tree, J48, RFT, MLP, and NB and gets the highest accuracy in MLP which is 98.53 and found a high decrease in False Positive. The drawback of this study is that the study should be evaluated using updated datasets.

Author [15] provides a thorough understanding of numerous algorithms and their hybridization, as well as insights into analyzing network algorithm literature and creating hybrid models with a wide range of metrics. This paper used the 'KDD'99 datasets & utilized GA & SVM algorithms & he found an accuracy of 98.33 by hybridization of both GA and SVM and found a decrease in FPR. The drawback of this study is that it should be evaluated using updated datasets like NSL-KDD and more.

Author [16] reveals that when applied to the Kyoto 2006+ dataset, the majority of machine learning algorithms perform admirably, with precision, recall, and accuracy consistently greater than 90%. However, when evaluated using the Receiver Operating Curve (ROC) measure, it is clear that the RBF (Radial Basis Function) method surpasses the other 7 strategies. The work was concluded utilizing the Kyoto 2006+ dataset and found that RBF has the highest accuracy. One drawback of this research is that there is a low recall rate.

Author [17] proposed that the IKPDS method outperforms kNN and KPDS in terms of classification completion time while maintaining comparable classification accuracy and error rates across a variety of threats. This paper used the KNN algorithm and KDD-NSL datasets and he found an accuracy of 99.95 that is high in comparison to the prior study.

Table 2: Comparison of Recent Intrusion Detection Models from 2012 to 2022

Authors	Published Year	Datasets Used	Algorithm/Classifiers Used	Accuracy or Outcome
Y. Li, et al. [18]	2012	KDD Cup 1999	Support Vector Machine (SVM)	98.62%
S. Mukherjee, et al. [19]	2012	NSL - KDD	Naïve Bayes (NB)	97.78%
N. Farnaaz & M.A. Jabbar [20]	2016	NSL - KDD	Random Forest (RF) and J48	99.67%
M.C. Belavagi & B. Muniyal [21]	2016	NSL - KDD	SVM , GNB , RF , LR	75% , 79% , 99% , 84%
K. Atefi, et al. [15]	2016	KDD Cup 1999	GA SVM Hybrid (GA + SVM)	84.03% 94.80% 98.33%
A.S. Amira, et al. [14]	2017	NSL - KDD	BF Tree Naïve Bayes J48 Random Forest MLP	98.24% 84.75% 97.68% 98.34% 98.53%
B. Brao & K. Swathi [17]	2017	NSL - KDD	KNN	99.95%
V.Hajisalem & S. Babaie [22]	2018	NSL - KDD	Random Forest Decision Tree	95.32% 81.86%
M. Belouch, et al. [23]	2018	UNSW – NB15	RF, SVM, NB, DT	97.49% , 92.28% , 74.19% , 95.82%
Sara Mohammadi, et al. [24]	2019	KDD Cup 1999	DT	95.03%
Ameera S. Jaradat, et al. [25]	2021	CICIDS 2017	SVM, RProp, DT	97.35% , 95.35% , 98.38%
R. Tahri, et al. [26]	2022	UNSW –NB15	KNN , NB , SVM	93.33% , 95.55% , 97.77%
		NSL - KDD	SVM	97.29%

			NB	67.26%

CONCLUSION

Machine learning has transformed intrusion detection, with ensemble and hybrid classifiers proving to be the most successful in increasing predicted accuracy and detection rates. In this study, we show various research from 2012 to 2022 that indicate the usefulness of these classifiers in intrusion detection systems. The usage of numerous classifiers leads to increased attack detection accuracy. Despite progress, resolving false positives and false negatives is critical. Researchers are encouraged to look at approaches with high precision rates.

REFERENCES

[1] Axelsson, S. (2000). “Intrusion detection systems: A survey and taxonomy.” Chalmers University of Technology.

[2] Lunt, T. F. (1993). “A survey of intrusion detection techniques.” Computers & Security, 12(4), 405-418.

[3] Patcha, A., & Park, J. M. (2007). “An overview of anomaly detection techniques: Existing solutions and latest technological trends.” Computer networks, 51(12), 3448-3470.

[4] Zhu, X., & Goldberg, A. B. (2009). “Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning,” 3(1), 1-130

[5] Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). “Reinforcement learning: A survey. Journal of artificial intelligence research,” 4, 237-285.

[6] Bou-Harb, E., Debbabi, M., & Assi, C. (2014).” Intrusion detection systems: A taxonomy and survey.” ACM Computing Surveys (CSUR), 46(4), 55.

[7] Quinlan, J. R. (1986). “Induction of decision trees.” Machine learning, 1(1), 81-106.

[8] Breiman, L. (2001). “Random forests. Machine learning,” 45(1), 5-32.

[9] Cortes, C., & Vapnik, V. (1995). “Support-vector networks.” Machine learning, 20(3), 273-297.

[10] Cover, T., & Hart, P. (1967). “Nearest neighbor pattern classification.” IEEE transactions on information theory, 13(1), 21-27.

[11] Dietterich, T. G. (2000). “Ensemble methods in machine learning. In Multiple classifier systems” (pp. 1-15). Springer, Berlin, Heidelberg.

[12] Kunal and M. Dua, "Machine Learning Approach to IDS: A Comprehensive Review," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 117-121, doi: 10.1109/ICECA.2019.8822120.

[13] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” 2015 Mil. Commun. Inf. Syst. Conf., no. November, pp. 1–6, 2015.

[14] A. S. Amira, S. E. O. Hanafi, and A. E. Hassanien, “Comparison of classification techniques applied for network intrusion detection and classification,” J. Appl. Log., vol. 24, pp. 109–118, 2017, doi: 10.1016/j.jal.2016.11.018.

[15] K. Atefi, S. Yahya, A. Rezaei, and S. H. B. M. Hashim, “Anomaly detection based on profile signature in network using machine learning technique,” Proc. - 2016 IEEE Reg. 10 Symp. TENSYMP 2016, pp. 71–76, 2016, doi:

10.1109/TENCONSpring.2016.7519380.

[16] M. Zaman and C. H. Lung, “Evaluation of machine learning techniques for network intrusion detection,” IEEE/IFIP Netw. Oper. Manag. Symp. Cogn. Manag. a Cyber World, NOMS 2018, pp. 1–5, 2018, doi: 10.1109/NOMS.2018.8406212.

[17] B. Brao and K. Swathi, “Fast kNN Classifiers for Network Intrusion Detection System,” no. April, 2017, doi:10.17485/ijst/2017/v10i14/93690.

[18] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai,“An efficient intrusion detection system based on support vector machines and gradually feature removal method”. Expert Systems with Applications, 39(1), 424-430, 2012.

[19] S. Mukherjee andN. Sharma, “Intrusion detection using naive Bayes classifier with feature reduction”, Procedia Technology, 4, 119-128, 2012.

[20] Farnaaz, N., & Jabbar, M. A. (2016). “Random forest modeling for network intrusion detection system.” Procedia Computer Science, 89, 213- 217.

[21] Belavagi, M. C., & Muniyal, B. (2016). “Performance evaluation of supervised machine learning algorithms for intrusion detection.” Procedia Computer Science, 89, 117-123.

[22] V. Hajisalem and S. Babaie, “A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection,” Comput. Networks, vol. 136, pp. 37–50, 2018, doi: 10.1016/j.comnet.2018.02.028.

[23] M. Belouch, S. El Hadaj andM. Idhammad, “Performance evaluation of intrusion detection based on machine learning using Apache Spark”, Procedia Computer Science, 127, 1-6, 2018.

[24] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, “Cyber intrusion detection by combined feature selection algorithm”, Journal of information security and applications, 44, 80-88, 2019.

[25] Jaradat, A. S., Barhoush, M. M., & Easa, R. S. B. (2022). “Network intrusion detection system: machine learning approach.” In Indonesian Journal of Electrical Engineering and Computer Science (Vol. 25, Issue 2, p. 1151). Institute of Advanced Engineering and Science. https://doi.org/10.11591/ijeecs.v25.i2.pp1151-1158

[26] Tahri, R., Balouki, Y., Jarrar, A., & Lasbahani, A. (2022).“Intrusion Detection System Using machine learning Algorithms.” In M. Sbihi, A. Mounadi, & M. Garoum (Eds.), ITM Web of Conferences (Vol. 46, p. 02003). EDP Sciences. https://doi.org/10.1051/itmconf/20224602003