Comparative Analysis of Machine Learning Models for Cloud Network Intrusion Detection

Muhammad Hamid Sikandar Hayat Aoun Muhammad Umar Fayyaz Sehrish Raza
hamidcit14@gmail.com, sh363525@gmail.com, aoun.muhammad@iub.edu.pk, umar.fayyaz@iub.edu.pk, sehrishraza6@gmail.com
Faculty of Engineering and Technology
The Islamia University of Bahawalpur, Pakistan

***Abstract*—Due to such rapid advancement in the field of cloud computing, the need has drastically emerged for much-improved security protocols in order to save it from nefarious cyber attacks. This would also entail understanding that, in such a dynamic setting, the role of Intrusion Detection Systems is imperative, acting as a sort of support system that helps in identifying malicious activities and intrusions. The paper explains the discussion of various machine learning techniques employed for intrusion detection in cloud systems. We test a wide range of models that use Random Forest, XGBoost, DNN, Decision Trees, KNN, Logistic Regression, SVM, and Naive Bayes to check their efficiencies for identifying various forms of assaults. This research paper also defines the respective pros and cons for each one of them. The following analysis makes its conclusion based on critical performance factors such as accuracy, recall, and F1-Score, which would be fundamental in determining the efficiency of such models in handling the forms of current network attacks. It is clear from the results that it is important to note that ensemble techniques such as XGBoost and RF models are superior to other models evaluated for.These models effectively balance the trade-off between high detection accuracy and low false positive rates, making the most robust choices for multi-class, complex cloud security systems.**

***Index Terms*—Intrusion Detection System, Machine Learning, UNSW-NB15, Cybersecurity, Comparative Analysis, Network Security.**

I. Introduction

Although cloud computing is a significant component in the growth of the planet, it is still faced with many challenges in terms of security threats. The cyber attack is directed at the cloud infrastructure, which is intended to affect the system. In relation to sophisticated cloud attacks, the use of signature based methods within the intrusion detection systems is not adequate. This is particularly a serious issue regarding the observation of control over threats on the network, which are becoming more common. The cloud acceptance has, unexpectedly, given a paradigm shift in the basic dynamics of implementation and management of IT infrastructure by organizations. With businesses migrating towards multi cloud systems, which use multiple cloud service providers within a short period of time, businesses get facilitated with advantages of redundancy, flexibility, and freedom from lock-in. It is henceforth advisable that highly sophisticated systems for the detection of potential threats should be developed for multi cloud systems [1] [2]. The most popular types of network attack are DDoS, user to root (U2R), remote to local (R2L), and network probing attacks.The aim of this research is to discuss the phenomenon of which Machine Learning Model is most efficacious in cloud NIDS so that it provides protection from contemporary Cyber threats. This decides which is the most accurate in detecting malicious traffics that are used contemporarily. Only models that are most generalized, providing real time solutions, which cancel trade offs in a cloud environment, are considered here. In order to examine the availability of reliable support from Machine Learning based Infiltration Detection systems in multi-cloud systems, we used the UNSW-NB15 Dataset. This particular dataset is considered a benchmark in the realm of cyber research, owing to the fact that it has sufficient detail regarding the amount of traffic that a researcher requires in order to make the distinction between normal network activity and malicious activity.The core of this research is the comparison of eight different models, from the state of the art Deep Neural Network (DNN), ensemble models XGBoost, Random Forest, to more classical models Naive Bayes, and SVM. Not only are detection rates assessed on how well we performed, but a variety of precision, recall, and F1 scores are considered as well. We affirm that, at the current state of affairs, the most fitting technique is the tree based ensemble technique, which is proven to be extremely accurate with a significantly reduced need for computational power, as is the case with deep learning models. Ultimately, this research is a step in the right direction in making multi cloud infrastructure a safer environment.

Objectives. Comparison of core classifiers: We want to use the UNSW-NB15 dataset to put through eight different machine learning methods in complete, thorough stress tests. These include algorithms such as SVM, DNN, XGBoost, and Random Forest. The idea here would be to set a clear baseline against which performance is measured for finding intrusions in complex cloud network traffic.

Architectural Face-Off: Here, we will compare directly how various Ensemble Learning methods like XGBoost do their job against deep learning architectures, such as DNN and regular linear classifiers. Given that, it should be possible to find out what type of architectural philosophy yields the most accurate detection rate for modern attack vectors. We will know the actual trade offs between complexity and model cost. That will help us find a good balance between cost and power. We want to learn which models can actually be deployed in cloud settings because of the limited resources by keeping an eye on how long the training takes and the time taken to do predictions.

Champion Mode Selection We want to find the fittest model with respect to multi cloud security using real world data. It would be our data that decided which one to use, and F1 Score along with FPR was considered as critical elements toward the determination of a safe yet feasible solution.

II. LITERATURE REVIEW

Developing Cloud-Based Intrusion Detection Cloud computing has made static security measures inadequate. Dynamic systems need adaptable solutions [3]. Traditional signature-based Intrusion Detection Systems (IDS) struggle against sophisticated and scalable cloud assaults. An anomaly-based detection approach employing ML/DL approaches has emerged to identify known and undiscovered threats [4]–[6] due to limitations.Increased complexity in multi-cloud setups requires NIDS that can operate in virtualized settings and reduce false

© 2026 TechShield Publications
DOI: https://doi.org/10.67535/tsp.000002.017
Published under the ISAISS 2026 Proceedings

positives [7]. Due to resource allocation volatility, network topology cannot be monitored statically [8].

A. *Battlefield Benchmarking*

Benchmark dataset quality determines ML-based IDS efficacy. Current research centers on three datasets:

1) UNSW-NB15 database: developed to replace KDD99, has 2.5 million records, including fuzzers, backdoors, and reconnaissance vulnerabilities [9], [10]. AL Masoodi (2024) achieved 82.79% validation accuracy with a 1D CNN trained on UNSW-NB15 for multi-cloud setups [11].

2) KDD99, NSL-KDD : eliminates bias and duplication for a cleaner testbed [12]. Random Forest models with 99.83% accuracy demonstrate the efficacy of tree-based classifiers [13], [14].

3) CICIDS-2017: simulates multiple attack profiles with realistic background traffic [15]. While Deep Autoencoder ANN achieved 98.7% accuracy, Random Forest obtained 99.5/NSL-KDD excels in multi-class classification but lacks current attack vectors compared to UNSW-NB15 [16].

B. *Cloud-Based Machine Learning*

Classic Algorithm Dominance: Despite neural network breakthroughs, classic ML algorithms, mainly ensemble approaches like Random Forest (RF), remain competitive. RF achieves above 99.5% accuracy in many experiments [16], [17] by successfully handling noisy features and preventing overfitting. While SVMs have high accuracy (98.97%) [14], their computational cost restricts their real-time cloud applicability [18]. Baseline models like Logistic Regression and Naive Bayes often struggle with complicated cyber attack patterns due to linearity and independence assumptions [13], [19]. Compared studies show that model complexity does not ensure superiority. According to [13], [14], [20], optimized ensemble trees (RF, J48) show better detection accuracy than deep learning models while preserving computational efficiency.

C. *Identification of Gaps & Our Contribution*

In spite of the scope of ongoing research, there are still certain critical gaps in it.

1) The Multi Cloud Blind Spot: Most literature on the subject is generalized on overall cloud infrastructure, such as single server installations, which overlook a multi cloud system's nuanced behavior [11].

2) Real World Friction: There is a lack of research that establishes a high degree of accuracy with an affordable computational cost of implementation [7]. "A system that detects 100% of the attacks but brings the network to a crawl is of no use."

3) Contribution: The contribution of this study lies in a systematic empirical comparison eight ML algorithm on the UNSW-NB15 dataset.It also comprehensively explores the trade off that exists between efficiency in detection and computational complexity in order to provide a practical solution approach for securing multi cloud systems.

D. *Theoretical*

Our design is formed by the Theory of Anomaly Detection by Denning [21], who proposed that there is a statistical pattern of malicious behavior that is distinct from the pattern of normal behavior. This is supplemented by the principles of Breiman for learning with ensembles [22] for variance reduction, with limitations imposed by the defense-in-depth approach by the Cloud Security Alliance [23].

III. METHODOLOGY

Our approach is founded on a sole hypothesis that the randomness of multi cloud traffic requires an adaptive defense mechanism that outpaces the attacking forces arrayed against it. Having noticed that conventional detection techniques tend to falter at the presence of complex and dynamic attack patterns, we resorted to a thorough relative analysis of state of the art Machine Learning solutions to develop a more robust Intrusion Detection System. Data Acquisition and Engineering The core of our experiment is the UNSW-NB15 data set. It is one of the few data sets that offers a realistic representation of the network traffic that is currently observed on the Internet, with a level of realism that even past network traffic data is not sophisticated enough to record. Real world data is never necessarily "model ready," however. In order to prepare the battlefield, we established a harsh pre processing challenge .Encoding: Using a similar process, the categorical variables about the protocols were encoded into binary forms that the computer could understand. Dealing with Class Imbalance: Noticing that attacks are overshadowed by a substantial amount of regular traffic, we applied the Synthetic Minority Over sampling Technique (SMOTE). .

A. *The Model Selection Strategy Rather than betting on a single algorithm,*

we deployed eight distinct classifiers to see which logic prevailed. Our selection spans the full spectrum of machine learning:

1) Ensemble Methods: Random Forest, XGBoost.

2) Deep Learning: Deep Neural Networks (DNN).

3) Traditional Classifiers: Decision Tree, KNN, Logistic Regression, SVM, and Naive Bayes.

This then allows us to investigate the degree to which various mathematical ideologies, such as the complex stratification of deep learning or the iterative boost of ensembles, solve for the extraction of subtle patterns in network noise.

B. *Training and Validation Protocol*

To ensure our findings hold up in the practical, we enforced a strict 80/20 split, dedicating 80% of the data to training and withholding 20% for validation.This study selected the dataset UNSW-NB15 because it has high volume and complexity for training to maximize performance.this dataset has 49 features and 9 attack categories like reconnaissance,backdoor,DoS,exploits,analysis,fuzzers, worms, shellcode and also contain normal traffic pattern.These things make dataset modern,comprehensive and publically available.

1) Data Preprocessing: The data preprocessing is done using ,feature selection (top 20 most important features) ,encoding,Normalization and splitting of data. Class Imbalance Handling using the SMOTE

C. *Overview of the dataset.*

The UNSW-NB15 benchmark in cybersecurity research captures the complexity of actual network traffic, rather than cleaned or theoretical models [24], [25]. The dataset includes typical network behavior and nine cyber-attack families, providing a comprehensive perspective of present threats [26], [27].

Fuzzer is an automated stress-testing tool that randomly injects faulty data streams to identify vulnerabilities [28].

Reconnaissance involves port scanning, traffic sniffing, and probing to identify network vulnerabilities for exploitation [28], [29].

Using backdoors in software or legal code allows remote unauthorized access by bypassing authentication [30], [31].

Denial of Service (DoS) describes attacks that drain resources, rendering systems inaccessible to authorized users [32] [33].

Exploits exploit software or hardware defects to gain control or privileges [34].

Algorithm-independent generic attacks include collisions in hash functions and mathematical properties [35].

Shell code is brief code that opens a command shell for quick system access during a target assault [36].

Worms are viruses that automatically spread across networks by exploiting vulnerabilities [37].

D. Model Architectures

When we were looking for the best multi-cloud guardian, we didn't restrict ourselves to a single approach, the author says.we cast a broad net and evaluate eight alternative algorithms that embody these three philosophies(Ensemble Learning, the intricacy of Deep Learning, and the statistical rigor of Traditional Classifiers)

The Power of the Crowd An Overview of Ensemble Methods The Random Forest and XGBoost algorithms are the most important ones in our research. Ensemble techniques are based on the premise that a 'committee' of weak learners, often known as decision trees, is more intelligent than any one person.

Consider the Random Forest algorithm, also known as the Parallel Approach, to be an engine that utilizes the "wisdom of the crowd. Through a technique known as bagging, it constructs a large number of decision trees that are fully independent of one another among themselves.it is very stable and resistant to being overfit [38].

The Sequential Approach, also known as Extreme Gradient Boosting (XGBoost), is extremely effective at catching the non-linear feature interactions that are necessary for identifying subtle assaults such as analysis exploits or fuzzers [39].

E. Machine Learning Classifiers

This work presents a comparison of eight machine learning methods, ranging from statistical ones to deep learning architecture.

1) K-nearest Neighbors: This is a non-parametric method that classifies the data points by the majority class of their nearest neighbors.

2) Decision Tree: A model that is based on rules, whereby there is a separation of data using "if-else" to gain maximum information. It has the structure of a tree.

3) Random Forest: A large ensemble of top-level trees that combines their output to boost the performances and reduce overfitting.

4) Support vector machines : find the optimal N-dimensional hyperplane for classification in high-dimensional issues.

5) Naive Bayes: This Bayes' theorem-based probability classifier assumes independent features to predict class probability.

6) Logistic regression: calculates class membership probabilities in binary class issues using logistic or sigmoid functions.

7) Extreme Gradient Boosting : It is a scalable and efficient gradient boosting technique used for creating sequential trees. It is employed for fixing weaknesses of models. It is appropriate for binary classification problems. It supports multi

8) Deep Neural Network:: This is a complex form of an artificial neural network with numerous hidden layers ranging from the input to output layers to handle complicated nonlinear events.

F. Evaluation Metrics

For the performance evaluation of the selected models, the following metrics were considered:

1) Accuracy: The quotient of accurately anticipated observations to the total number of observations is called accuracy.

2) Precision: is the ratio of correctly predicted positive observations to the total number of positively forecasted observations.

3) The confusion matrix: is a tabular summary of the performance of the classification model through the use of True Positives, True Negatives, False Positives, and False Negatives. It is a resampling process used to validate the model on a restricted data sample.

4) 4-Fold Cross-Validation: The information is divided into four subsets, and the model is trained on three of them before being tested on the fourth subset. This process is repeated four times to guarantee that the model is resilient.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1-Score	Training Time
XGBoost	**0.970**	**0.970**	**0.970**	**0.970**	**45.2 s**
Random Forest	0.960	0.960	0.960	0.960	38.7 s
DNN	0.945	0.945	0.945	0.945	156.8 s
Decision Tree	0.920	0.920	0.920	0.920	12.3 s
KNN	0.900	0.900	0.900	0.900	8.4 s
SVM	0.830	0.830	0.830	0.830	89.3 s
Logistic Regression	0.800	0.800	0.800	0.800	15.6 s
Naive Bayes	0.750	0.700	0.750	0.750	3.2 s
				Total Training Time	369.5 s

5) Precision-recall curves: are a data visualization tool showing the trade-off between accuracy and recall at a variety of thresholds. The training time represents the computation time required for the training of the model and is a measure of the efficiency with which cloud deployment is being done.

G. Computational Environment and Reproducibility

Experiments were conducted on a system with Core (i7) ,(16) GB of RAM, a (4) GB NVIDIA GPU, and a (320) GB NVMe SSD. Model training was performed in Python (3.9.13) using TensorFlow (2.12.0), PyTorch (2.9.0), scikit-learn (1.2.2), and XGBoost (1.7.5) and NumPy (1.23.5). To ensure reproducibility, a fixed random seed of 42 was applied across all models, with `PYTHONHASHSEED=42` and `TF_DETERMINISTIC_OPS=1` enabled.

H. Key model hyperparameters

In the XGBoost model, n estimators were set to 200, subsample=0.8, max_depth=6, and colsample_bytree=0.8. Random Forest classifier was used with n estimators = 100, max_depth=None, min_samples_split=2, and max_features=sqrt. For Decision Tree, the criterion used was gini with max_depth=10, and min_samples_split=2. DNN, the architecture consisted of 64, 32, and 10 neurons in the hidden layers with ReLU activation and Softmax in the output layer, optimized by Adam with a batch size of 32 over 15 epochs. In SVM, RBF kernel was used with C = 1.0, γ = scale, and max_iter = 2000. K-Nearest Neighbors classifier was instantiated with n_neighbors = 5, metric = 'minkowski', and leaf_size = 30. Logistic Regression, ℓ_2 regularization with C = 1.0, lbfgs solver, and multinomial classification was applied, whereas Naive Bayes used a variance smoothing value of 1×10^{-9}.

IV. EXPERIMENTAL RESULTS & ANALYSIS

1) Comparative Analysis of Performance : The comprehensive summary of the performance metrics of all the machine learning models applied are given in Table 1.

2) Best Performer: Overall, XGBoost turned out to be the better model since it achieved the highest scores with respect to all important parameters, with an accuracy, precision, recall, and F1-score of 97.0%.It is the best option with regard to finding complicated attack routes in a setting that includes several clouds because of its ability to handle non-linear feature interactions and to resist overfitting.

3) Strong Competitors: Random Forest and DNN were ranked second, with an accuracy of 96.0%, hence making it the second-best performance in the competition. Although that sounds impressive, the fact that it was able to achieve this level with less time spent training-38.7 seconds compared to 45.2 seconds by XGBoost-suggests that it is a very efficient choice for systems in which training speed is considered. Besides, the DNN exhibited excellent detection capabilities with an accuracy rate of 94.5 percent. Despite this, it took the DNN a whopping 156.8 seconds to train roughly four times longer than the Random Forest model.

Fig. 1. XGBoost Confusion Matrix

4) Baseline performance : was measured using support vector machines, logistic regression, and naive bayes. The Support Vector Machine (SVM) and Logistic Regression both attained an accuracy of 83.0% and 80.0%, respectively. With an accuracy of 75%, the Naive Bayes algorithm demonstrated the least amount of performance.

5) Assessment of the Training Time : On the basis of the findings obtained from this research, XGBoost is one of the most precise approaches, but it consumes a lot of processing power. In terms of 'bang for a buck' compared to XGBoost, Random Forest is the most optimal because it results in close to the best possible precision simultaneously facilitating faster training. At the other end, when precision is considered, Naive Bayes is highly inefficient because it is extremely fast but results in inaccurate precision that is not acceptable.

A. *Confusion Matrix Analysis Discussion*

1) Random Forest / XGBoost Ensemble Methods: The ensemble methods had the highest density along the diagonal indicating a higher predictive accuracy. XGBoost (Figure 1) XGBoost had the highest True Positive (TP) of the majority class which was 7,146 Normal traffic injections. It also classified 3,628 instances correctly with very few False Positives and thus was very accurate in detecting Generic attacks. The wrong detection of Exploits and DoS attacks (315) was a slight drawback. Besides other models, XGBoost was able to lessen this misclassification more. Random Forest :Like XGBoost, Random Forest also classified 7,100 Normal examples.

2) Deep Learning (DNN) : The DNN exhibited excellent generalization but showed certain limitations related to its structure in the recognition of spectrally similar attacks. Traffic Confusion: The model was able to identify 7,050 Normal instances accurately, which is very close to the count obtained by the tree-based ensembles but still lower. Pattern Overlap: A unique error pattern showed up where 340 Exploits were incorrectly labeled as DoS and 78 Exploits were incorrectly labeled as Fuzzers.

3) Decision Tree: The single Decision Tree presented the highest overall accuracy (92%) but at the same time had more "noise" in its off-diagonal predictions than Random Forest did.Overfitting Evidence: It mistakenly categorized 400 Exploits as DoS, which corresponds to a significantly higher error rate than the ensemble methods.

4) KNN & SVM: Both the models had a tough time separating high-dimensional features, which showed that simple distance or margin-based classifiers are not enough for modern intrusion detection.SVM Leakage: The SVM classifier was tremendously leaky, misclassifying 460 cases of Exploits as the DoS type and 110 Normal cases as the DoS type.

5) Logistic Regression: As an instance of a linear model, Logistic Regression, to put it another way, did not indeed capture

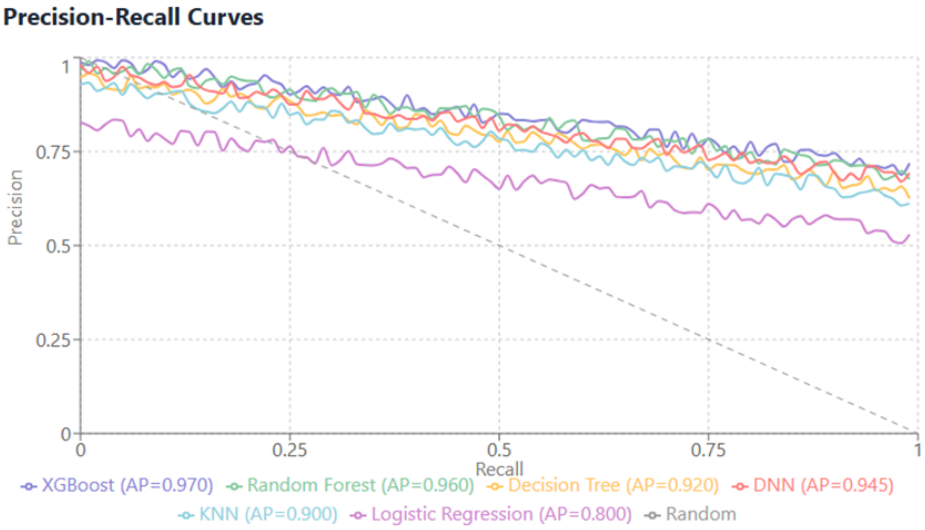

Fig. 2. PR CURVE

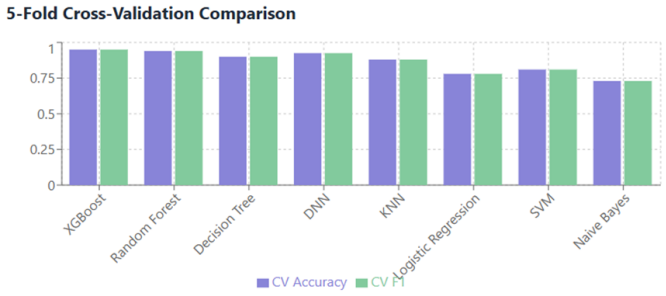

Fig. 3. 5-Fold Cross-Validation

the complex relationships among the classes and was, therefore, responsible for the widespread confusion.Critical Failures It included among the 480 cases of Exploits misclassified as DoS and presented a very high rate of False Negatives for Worms and Shellcode that was a cause for concern.

6) The Baseline Failure: Naive Bayes Naive Bayes : The confusion matrix for Naive Bayes gives a confirmation of the reason for the lowest accuracy.The diagonal is considerably weakened compared to all the other models. The most striking thing is that Normal traffic was often confused with DoS and Fuzzers (280 and 110 samples, respectively). This is a massive error, showing that the attributes of the UNSW-NB15 dataset (for instance, ”protocol,” ”duration,” and ”packet count”) are highly dependent. Conclusion on Matrix Analysis: The confusion matrices confirms that XGBoost is the most reliable layer of security, followed by Random Forest.They not only achieve the highest accuracy but also, and more importantly, reduce the number of False Negatives for critical stealthy attacks.

B. Discussion on Precision-Recall Analysis

PR Curve Analysis Figure 2 High-Precision Stability—XGBoost, RF:RF and XGBoost plot quite similarly.Horizontal lines around y = 1.0 at various recall levels.For Recall values at approximately 1.0.XGBoost outperformed and outclassed LR by an amazing difference at 0.970 and RF was also very impressive, standing at 0.960 compared to LR’s 0.800.

C. Cross Validation

A Discussion on Five-Fold Cross Validation Cross Validation was performed in order to assess the statistical reliability of our results and to ensure that the declared accuracy was not due to an artifact of a particular data split. In this rigorous validation method, the dataset is split into five distinct subsets. The models are trained and tested in rotating subsets to ensure that each data point is reviewed. Figure 3 visually represents the average CV accuracy in purple and the average CV F1-Score in green for each of the eight models.

V. CONCLUSIONS

The study of eight machine learning algorithms on the UNSW-NB15 dataset indicates notable disparities in performance for multi-cloud Intrusion Detection Systems (IDS). Ensemble approaches captured complicated, non-linear network traffic patterns better than single learners and traditional statistical models. The best method was XGBoost, which combined excellent accuracy with gradient boosting for high-dimensional interactions. Random Forest performed reliably throughout validation folds with reduced computational cost. The Naive Bayes was the worst because it assumes feature independence, which is untrue for network packet properties connected with one another. Deep Neural Networks are very accurate but are computationally expensive to train. Hence, they are inappropriate for real-time IDS implementation. This study reveals that the deep models, while accurate, present the best balance of predictive performance, computational efficiency, and reliability with tree-based ensemble methods like XGBoost and Random Forest, therefore being ideal in protecting cloud infrastructures against evolving cyber threats.

REFERENCES

[1] A. Vinolia, N. Kanya, and V. N. Rajavarman. Machine learning and deep learning based intrusion detection in cloud environment: A review. In *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 952–960, 2023.

[2] S. Devi and D. A. K. Sharma. Understanding of intrusion detection system for cloud computing with networking system. *International Journal of Computer Science and Mobile Computing*, 9(3):19–25, 2020.

[3] S. D. Muller, S. R. Holm, and J. Sondergaard. Benefits of cloud computing: Literature review in a maturity model perspective. *Communications of the Association for Information Systems*, 37, 2015.

[4] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013.

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

[6] S. Paul, Y. Jain, M. Samaka, and J. Pan. Application delivery in multi-cloud environments using software defined networking. *Computer Networks*, 68:166–186, 2014.

[7] M. D. Sreeramulu, K. Suganyadevi, A. R. Neravetla, and A. S. Mohammed. Network intrusion detection in cloud computing environments. In *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)*, 2024.

[8] A. B. Nassif, M. A. Talib, Q. Nassir, H. Albadani, and F. D. Albab. Machine learning for cloud security: A systematic review. *IEEE Access*, 2021.

[9] N. Moustafa and J. Slay. Unsw-nb15: A comprehensive data set for network intrusion detection systems. In *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015.

[10] N. Moustafa and J. Slay. Evaluation of network anomaly detection systems. *Information Security Journal: A Global Perspective*, 2016.

[11] H. J. K. Al Masoodi. Evaluating the effectiveness of machine learning-based intrusion detection in multi-cloud environments. *Babylonian Journal of Internet of Things*, pages 94–105, 2024.

[12] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.

[13] A. M. Mahfouz, D. Venugopal, and S. G. Shiva. Comparative analysis of machine learning classifiers for network intrusion detection. In *Conference Paper*, 2019.

[14] S. et al. Konkimalla. A comparative analysis of network intrusion detection using different machine learning techniques. *Journal of Contemporary Education Theory & Artificial Intelligence*, 2023.

[15] I. Sharafaldin, A. H. Lashkari, and A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy*, 2018.

[16] S. M. Kasongo and Y. Sun. Performance analysis of intrusion detection systems using feature selection on unsw-nb15. *Journal of Big Data*, 2020.

[17] J. Akoto and T. Salman. Machine learning vs deep learning for anomaly detection in multi-cloud environments. In *IEEE Cloud Summit*, 2022.

[18] R. K. Vigneswaran, R. Vinayakumar, K. Soman, and P. Poornachandran. Evaluating shallow and deep neural networks for network intrusion detection. In *ICCCNT*, 2018.

[19] A. Ng and M. I. Jordan. On discriminative vs generative classifiers. *Advances in Neural Information Processing Systems*, 15, 2002.

[20] Q. O. Ahmed. Machine learning for intrusion detection in cloud environments. *Journal of Artificial Intelligence General Science*, 2024.

[21] D. E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, 1987.

[22] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[23] Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing, 2021.

[24] P. S. Negi, A. Garg, and R. Lal. Intrusion detection and prevention using honeypot network for cloud security. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 129–132, 2020.

[25] B. T. Devi, S. Shitharth, and M. A. Jabbar. An appraisal over intrusion detection systems in cloud computing security attacks. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020.

[26] M. Kumar and A. K. Singh. Distributed intrusion detection system using blockchain and cloud computing infrastructure. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 248–252, 2020.

[27] A. Guezzaz, A. Asimi, Y. Asimi, M. Azrour, and S. Benkirane. A distributed intrusion detection approach based on machine learning techniques for cloud security. In *Intelligent Systems in Big Data, Semantic Web and Machine Learning*, pages 85–94. Springer, 2021.

[28] N. D. Patel, B. M. Mehtre, and R. Wankar. Od-ids2022: Generating a new offensive defensive intrusion detection dataset. *International Journal of Information Technology*, 15:4349–4363, 2023.

[29] M. G. Raj and S. K. Pani. A meta-analytic review of intelligent intrusion detection techniques in cloud computing environment. *International Journal of Advanced Computer Science and Applications*, 12, 2021.

[30] A. S. Iliyasu and H. Deng. N-gan: A novel anomaly-based network intrusion detection with generative adversarial networks. *International Journal of Information Technology*, 14:3365–3375, 2022.

[31] S. P. U. Kirana and D. A. D'Mello. Energy-efficient enhanced particle swarm optimization for virtual machine consolidation in cloud environment. *International Journal of Information Technology*, 13:2153–2161, 2021.

[32] R. Keshri and D. P. Vidyarthi. Communication-aware, energy-efficient vm placement in cloud data center using ant colony optimization. *International Journal of Information Technology*, 15:4529–4535, 2023.

[33] K. Srinivas, N. Prasanth, and R. et al. Trivedi. A novel machine learning inspired algorithm to predict real-time network intrusions. *International Journal of Information Technology*, 14:3471–3480, 2022.

[34] J. Wei, C. Long, J. Li, and J. Zhao. An intrusion detection algorithm based on bag representation with ensemble support vector machine in cloud computing. *Concurrency and Computation: Practice and Experience*, 32(24):e5922, 2020.

[35] N. M. Ibrahim and A. Zainal. A distributed intrusion detection scheme for cloud computing. *International Journal of Distributed Systems and Technologies*, 11(1):68–82, 2020.

[36] O. Alkadi, N. Moustafa, B. Turnbull, and K. K. R. Choo. A deep blockchain framework-enabled collaborative intrusion detection for protecting iot and cloud networks. *IEEE Internet of Things Journal*, 8(12):9463–9472, 2021.

[37] S. Krishnaveni, S. Sivamohan, S. S. Sridhar, and S. Prabakaran. Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. *Cluster Computing*, 24(3):1761–1775, 2021.

[38] M. Alweshah, S. Alkhalaileh, M. Beseiso, M. Almiani, and S. Abdullah. Intrusion detection for iot based on a hybrid shuffled shepherd optimization algorithm. *The Journal of Supercomputing*, 78(10):12278–12309, 2022.

[39] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters. Intelligent dynamic malware detection using machine learning in ip reputation for forensics data analytics. *Future Generation Computer Systems*, 118:124–141, 2021.