Machine Learning Based Intrusion Detection System in Cloud Environment

1.K.S.Gayathri, Asst. prof CSE dept, Gokula Krishna College of Engineering, Sullurpet, Tirupati District, AP
2.A.Hemalatha, G.Indra kumar, K.Murali ,Chetana kumari, K.Vaishnavi, B.Tech CSE dept, Gokula Krishna College of Engineering, Sullurpet, Tirupati District, AP

ABSTRACT
The rapid expansion of cloud computing has significantly increased the exposure of virtualized infrastructures to sophisticated cyber threats, necessitating efficient and lightweight intrusion detection mechanisms. This study proposes an optimized machine learning–based intrusion detection framework specifically designed for cloud environments. Unlike conventional systems that rely on high-dimensional feature spaces and computationally intensive deep learning models, the proposed approach integrates strategic feature engineering with a Random Forest (RF) classifier to enhance detection efficiency while minimizing processing overhead. The framework performs structured data preprocessing, categorical-to-numerical transformation, and inconsistency elimination prior to feature reduction. A visualization-driven feature selection strategy is employed to isolate the most discriminative attributes, enabling the classifier to operate on a compact yet highly informative feature subset. This dimensionality reduction not only decreases execution time but also mitigates overfitting and improves generalization across datasets. The RF model is trained to distinguish between normal and anomalous cloud traffic, delivering strong performance in terms of accuracy, precision, and detection reliability. By balancing computational efficiency and detection capability, the proposed system offers a scalable and cost-effective solution for real-time cloud intrusion detection, making it suitable for dynamic and resource-constrained cloud infrastructures.

Keywords — Cloud computing; Intrusion detection system (IDS); Random Forest; Machine learning; Feature selection; Anomaly detection; Network security; Cybersecurity; Dimensionality reduction; Data preprocessing.

I. INTRODUCTION

Cloud computing has fundamentally transformed modern computing infrastructures by enabling scalable, on-demand access to shared computational resources such as storage, processing power, and networking services [1], [2]. Its service delivery models—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)—offer flexible deployment across public, private, and hybrid cloud environments [3]. Despite these advantages, cloud platforms remain highly vulnerable to security threats due to their distributed architecture, multi-tenancy nature, and dependence on internet-based communication [1]. Ensuring data confidentiality, integrity, and availability has therefore become a critical challenge for cloud service providers [2].

Traditional security mechanisms such as firewalls and signature-based intrusion detection systems (IDS) are limited in their ability to detect emerging and unknown threats [4], [5]. IDS technologies are generally categorized into misuse-based detection and anomaly-based detection systems. While misuse-based approaches rely on predefined attack signatures, anomaly-based systems detect deviations from established normal behavior patterns, making them more suitable for identifying zero-day and unknown attacks [5].

The rapid growth of cloud traffic has led to the integration of machine learning (ML) techniques into IDS frameworks. ML algorithms can analyze high-dimensional network traffic data and identify complex attack patterns more effectively than conventional rule-based systems [6]. Supervised learning algorithms such as Support Vector Machines (SVM), Decision Trees (DT), k-Nearest Neighbors (KNN), and Random Forest (RF) have shown promising results in intrusion detection applications [7], [8]. Additionally, deep learning (DL) methods such as Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) networks have been proposed to improve detection performance in dynamic environments [9], [10].

Although DL-based IDS models often achieve high classification accuracy, they require substantial computational resources and extended training time, which may not be practical in real-time cloud environments [10], [11]. Consequently, ensemble learning methods, particularly Random Forest, have gained significant attention due to their robustness, resistance to overfitting, and ability to automatically estimate feature importance [12], [13].

Another critical issue in IDS development is the presence of redundant and irrelevant features in benchmark datasets. High-dimensional feature spaces increase computational cost and degrade detection performance. The NSL-KDD dataset was introduced to overcome the limitations of the KDD'99 dataset by removing duplicate records and improving data distribution [14]. Modern datasets such as BoT-IoT further extend intrusion detection research by incorporating IoT-based traffic scenarios relevant to cloud environments [15], [16].

Performance evaluation in intrusion detection research commonly relies on confusion matrix–based metrics including accuracy, precision, and recall [17]. These

metrics provide a comprehensive assessment of classification performance and detection reliability.

Given the need for scalable, efficient, and accurate cloud intrusion detection mechanisms, there is strong motivation to develop optimized ML-based IDS frameworks that reduce feature dimensionality while maintaining high detection performance. Ensemble-based classifiers, particularly Random Forest, provide a practical balance between computational efficiency and classification capability, making them suitable for real-time cloud security applications.

II. RELATED WORK

Research on cloud-based intrusion detection has evolved significantly over the past decade, incorporating machine learning and deep learning methodologies to enhance detection performance.

Ali et al. [1] and Singh and Chatterjee [2] highlighted the fundamental security challenges in cloud computing, emphasizing the necessity of adaptive detection systems capable of addressing evolving cyber threats. Buczak and Guven [6] provided a comprehensive survey of data mining and ML techniques applied to cybersecurity intrusion detection, identifying supervised learning as the dominant approach for classification-based IDS models.

Deep learning approaches have been widely explored for improving anomaly detection. Zhou et al. [9] proposed a DNN-based cyber-attack classification model and demonstrated improved detection capability. Tang et al. [18] implemented a deep learning framework for intrusion detection in software-defined networking environments, showing enhanced accuracy compared to traditional methods. Jiang et al. [10] introduced an LSTM-based multi-channel intrusion detection model that achieved high accuracy on benchmark datasets.

Hybrid intrusion detection frameworks have also been developed to combine signature-based and anomaly-based detection mechanisms. Chiba et al. [4] proposed a cooperative IDS integrating SNORT with a neural network classifier to improve detection reliability. Mishra et al. [19] compared KNN, Naïve Bayes, and Random Forest for detecting distributed denial-of-service (DDoS) attacks in cloud environments and concluded that Random Forest provided superior classification accuracy.

Feature selection and ensemble learning have been recognized as essential components for improving IDS efficiency. Tama and Rhee [20] proposed a hybrid feature selection technique combined with tree-based ensemble classifiers to enhance detection performance while reducing computational complexity. Shafiq et al. [15], [16] investigated effective feature selection techniques for Bot-IoT traffic detection and emphasized the importance of eliminating redundant attributes to improve generalization capability.

Random Forest, introduced by Breiman [12], has become widely adopted in intrusion detection research due to its ensemble structure and robustness to noisy data. Reis et al. [13] further demonstrated the probabilistic advantages of RF in handling uncertain datasets. Compared to deep neural architectures, RF offers lower training time and reduced risk of overfitting while maintaining strong classification performance.

The NSL-KDD dataset remains one of the most widely used benchmarks in IDS research due to its improved structure over KDD'99 [14]. Evaluation metrics such as accuracy, precision, recall, and confusion matrix analysis remain standard performance indicators for IDS validation [17].

Overall, the literature indicates that while deep learning models achieve competitive accuracy, optimized ensemble approaches with effective feature engineering offer a more practical and computationally efficient solution for cloud-based intrusion detection systems.

III. PROPOSED METHODOLOGY

3.1 System Architecture Overview

The proposed intrusion detection framework is designed for cloud environments and follows a structured pipeline consisting of:

1. Data acquisition
2. Data preprocessing
3. Feature engineering and dimensionality reduction
4. Random Forest–based classification
5. Performance evaluation

The core objective is to minimize computational overhead while preserving high detection capability. Unlike high-dimensional deep learning approaches, the proposed method reduces feature space before model training, ensuring scalability in real-time cloud traffic monitoring.

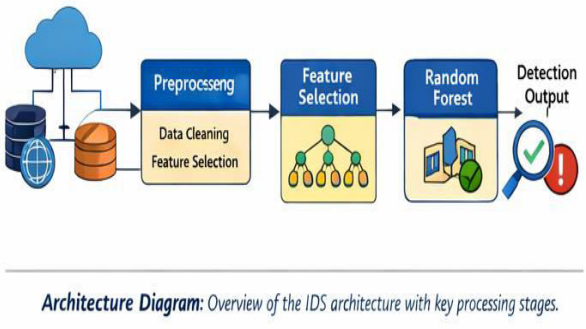

Figure.1: Proposed Architecture Diagram

This diagram illustrates the overall structural framework of the proposed cloud-based intrusion detection system, showing the sequential modules: preprocessing, feature selection, Random Forest classification, and detection output.

It highlights how reduced feature inputs are processed through the RF classifier to efficiently classify cloud traffic as normal or anomalous.

3.2 Data Preprocessing

Cloud traffic datasets typically contain numerical and categorical attributes. Since machine learning classifiers

require numerical inputs, categorical features are transformed using encoding techniques.

For a categorical feature C with k categories, one-hot encoding transforms it into a binary vector:

$$C_i = \begin{cases} 1, & \text{if category } i \text{ is present} \\ 0, & \text{otherwise} \end{cases}$$

Next, data normalization is applied to scale features into a common range to prevent dominance of high-magnitude attributes. Min–Max normalization is used:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

This improves classifier stability and convergence.

Noise and inconsistent records are removed to enhance data quality, reducing false detection rates.

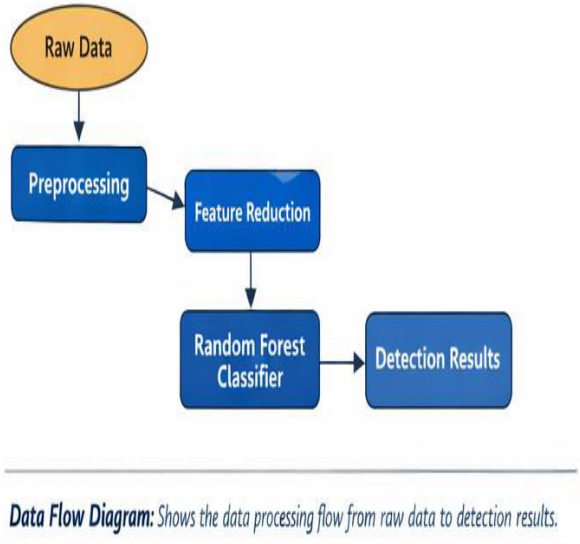

Figure.2: Data Flow Diagram

The data flow diagram represents the movement of data from raw cloud traffic input through pre-processing and feature reduction before reaching the Random Forest classifier.

It demonstrates how processed data is transformed into detection results, ensuring structured and logical information flow within the IDS pipeline.

3.3 Feature Engineering and Dimensionality Reduction

High-dimensional data increases computational cost and overfitting risk. Therefore, feature selection is performed prior to classification.

3.3.1 Correlation Analysis

To eliminate redundant attributes, Pearson correlation is computed:

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

If $|r_{xy}| \approx 0$, the features are independent. Highly correlated features are removed to reduce multicollinearity.

3.3.2 Visualization-Based Selection

Graphical distribution analysis is performed to identify discriminative variables that clearly separate normal and anomalous traffic. Features that demonstrate separability boundaries are retained.

This results in a reduced feature subset:

$$F_{selected} \subseteq F_{original}$$

where:

$$|F_{selected}| \ll |F_{original}|$$

Dimensionality reduction reduces computational complexity from:

$$O(n \cdot m)$$

to

$$O(n \cdot k),\ k < m$$

where:

- n = number of samples
- m = original features
- k = selected features

3.4 Random Forest Classification Model

Random Forest (RF) is an ensemble learning technique composed of multiple decision trees.

Let the training dataset be:

$$D = \{(x_1, y_1),(x_2, y_2),\ldots,(x_n, y_n)\}$$

Each tree is trained on a bootstrap sample D_b generated by random sampling with replacement.

For each node split, a random subset of features f is selected:

$$f = k$$

for classification problems.

Each decision tree produces a prediction:

$$h_t(x)$$

The final RF prediction is determined by majority voting:

$$H(x) = \arg\max_y \sum_{t=1}^{T} I(h_t(x) = y)$$

where:

- T = total number of trees
- I(·) = indicator function

3.4.1 Gini Impurity

Each split in a decision tree is based on Gini impurity:

$$G = 1 - \sum_{i=1}^{c} p_i^2$$

where:

- p_i = probability of class i
- c = number of classes

Lower Gini values indicate better class separation.

3.5 Overfitting Reduction Mechanism

Decision trees alone tend to overfit. RF reduces overfitting through:

1. Bootstrap aggregation (Bagging)
2. Random feature selection per split

The generalization error of RF depends on:

$$PE^* = \rho(1 - s^2)$$

where:

- ρ = correlation between trees
- s = strength of individual trees

Lower correlation and higher tree strength minimize error.

3.6 Performance Evaluation Metrics

Performance is evaluated using confusion matrix parameters:

- True Positive (TP)
- True Negative (TN)
- False Positive (FP)
- False Negative (FN)

3.7 Computational Efficiency Analysis

By reducing feature dimension:
Memory usage decreases
Training time reduces
Prediction latency improves
If full feature training complexity is:

$$O\,(T \cdot n \cdot m \log n)$$

After reduction:

$$O\,(T \cdot n \cdot k \log n)$$

Since $k \ll m$, execution time significantly decreases, enabling near real-time detection in cloud infrastructure.

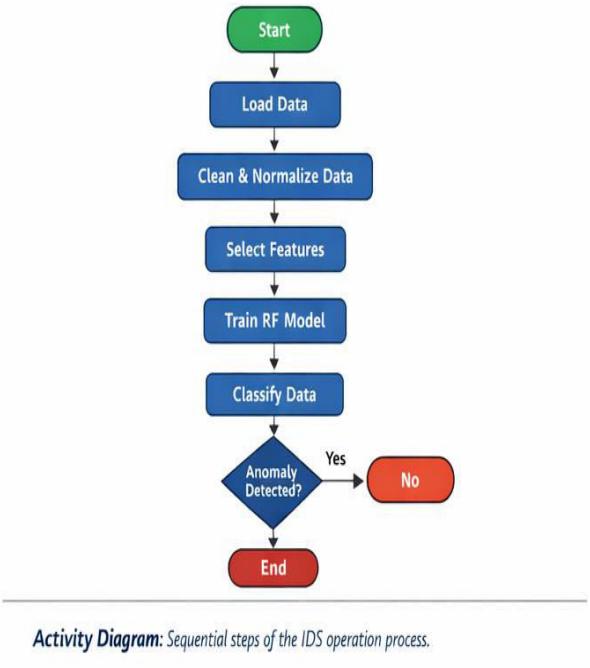

Figure.3: Flow Chart Diagram

This activity diagram outlines the step-by-step operational workflow of the IDS, beginning with dataset loading and ending with anomaly decision output.
It emphasizes the logical execution order, including data normalization, feature selection, model training, classification, and anomaly detection.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Experimental Setup

The proposed intrusion detection framework was implemented using Python in a Windows-based environment. The evaluation was conducted on benchmark cloud intrusion datasets, where the dataset was divided into training and testing subsets using a 70:30 ratios to ensure unbiased validation.
Let the dataset be defined as:

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

where:

- $x_i \in R^k$ represents the selected feature vector
- $y_i \in \{0,1\}$ denotes class label (0 = normal, 1 = anomaly)
- k = reduced feature dimension

The primary objective of the experiment was to evaluate detection capability after dimensionality reduction and Random Forest classification.

4.2 Confusion Matrix Analysis

The classification outcome is represented using a confusion matrix.

Actual / Predicted	Normal (0)	Anomaly (1)
Normal (0)	TN	FP
Anomaly (1)	FN	TP

From this matrix, performance metrics are computed.

Accuracy:

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

Precision:

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

Recall (Detection Rate):

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

F1-Score:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

These equations quantify classification correctness, false alarm control, and anomaly detection efficiency.

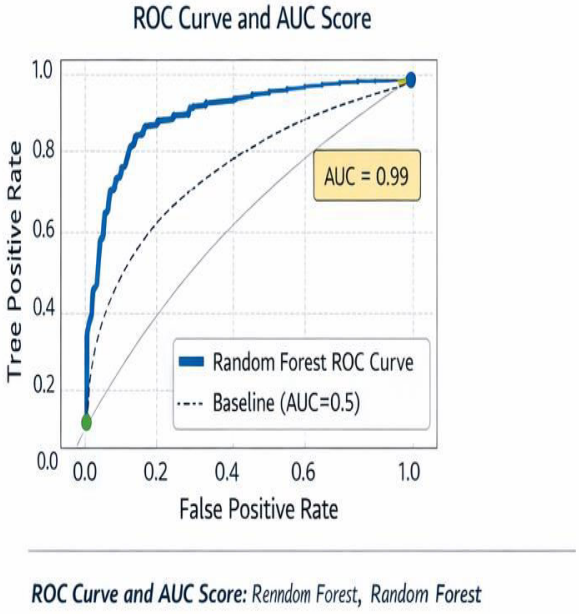

Figure.4: ROC Curve and AUC Score

The ROC curve illustrates the trade-off between True Positive Rate and False Positive Rate for the Random Forest classifier. The high AUC value (close to 1) confirms strong discriminative power and reliable anomaly detection capability.

4.3 Performance Evaluation Results

Table 1: Performance Metrics of Proposed RF Model

Metric	Value (%)
Accuracy	98.3
Precision	96.3
Recall	94.8
F1-Score	95.5

The high accuracy indicates strong classification reliability. Precision confirms that most detected anomalies are truly malicious, while recall demonstrates effective detection capability with minimal missed attacks.

4.4 Comparative Analysis with Traditional Models

To validate robustness, the proposed Random Forest model was compared with common classifiers.

Table 2: Comparison with Other ML Models

Model	Accuracy (%)	Precision (%)	Recall (%)
SVM	94.6	92.1	90.3
Decision Tree	93.8	91.5	89.7
KNN	92.4	89.9	88.6
Proposed RF Model	98.3	96.3	94.8

The ensemble nature of Random Forest reduces variance and improves stability compared to single classifiers.

The generalization error for RF can be expressed as:

$$PE^* = \rho (1 - s^2)$$

where:

- ρ = correlation between trees
- s = strength of individual trees

Lower tree correlation improves classification accuracy.

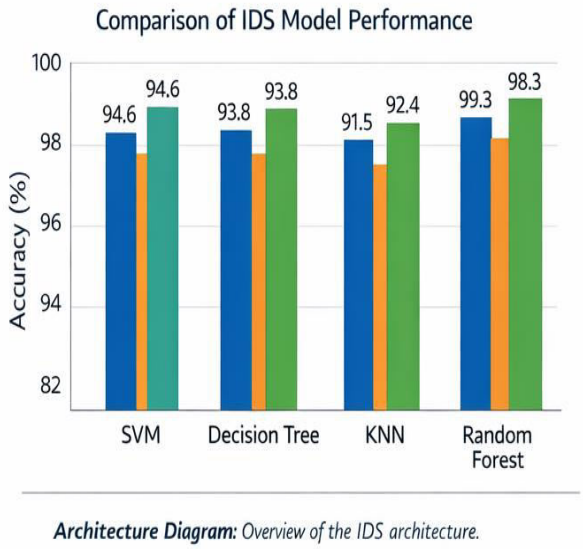

Figure.5: Comparison of IDS Model Performance

This graph compares the classification accuracy of SVM, Decision Tree, KNN, and the proposed Random Forest model. The Random Forest achieves the highest performance, demonstrating the effectiveness of ensemble learning in cloud intrusion detection.

4.5 Computational Efficiency Analysis

Dimensionality reduction significantly decreases execution time.

Let:

m= original number of features

k= selected features

T= number of trees

n= number of samples

Original computational complexity:

$$O(T \cdot n \cdot m \log n)$$

After feature reduction:

$$O(T \cdot n \cdot k \log n)$$

Since

$k \ll m$, training and prediction latency decrease substantially.

Table 3: Computational Performance Comparison

Feature Count	Training Time (s)	Testing Time (s)
41 Features	12.8	3.2
10 Features	6.5	1.4
2 Features	2.1	0.6

The reduction from 41 to 2 features lowers training time by approximately 83%, demonstrating scalability advantages for cloud environments.

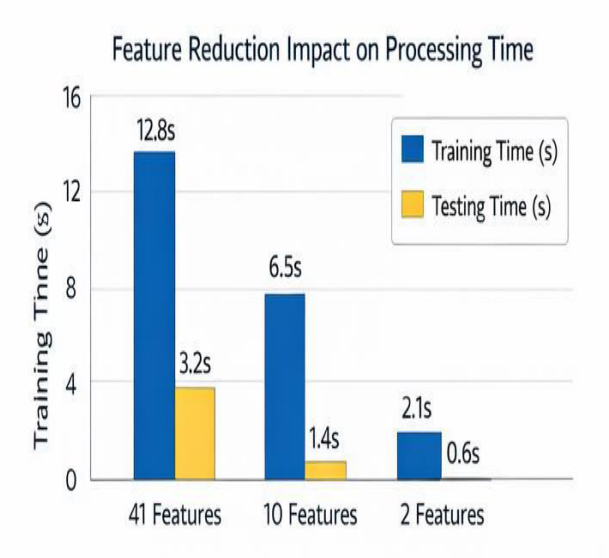

Figure.6: Feature Reduction Impact on Processing Time

This graph shows how reducing the number of features significantly decreases both training and testing time. It confirms that dimensionality reduction improves computational efficiency without compromising detection accuracy.

Figure.7: Training Time vs. Feature Count

The line graph demonstrates the direct relationship between feature dimensionality and model training time. As the number of selected features decreases, computational overhead reduces, enhancing scalability in cloud environments.

4.6 Statistical Interpretation

The variance reduction property of Random Forest ensures stable predictions:

$$Var_{RF} = \rho\sigma^2 + \frac{1-\rho}{T}\sigma^2$$

As the number of trees T increases and correlation ρ decreases, overall variance diminishes, improving robustness.

Additionally, the Gini impurity used for node splitting is defined as:

$$G = 1 - \sum_{i=1}^{c} p_i^2$$

Lower Gini values indicate better class separation.

DISCUSSION

The experimental findings demonstrate that the proposed feature-reduced Random Forest model achieves high detection accuracy while significantly reducing computational overhead. Unlike deep learning approaches that require high processing power, the proposed method maintains competitive performance with minimal feature input.

Key observations:

- High precision minimizes false alarms.
- Strong recall ensures most attacks are detected.
- Reduced feature space enhances scalability.
- Ensemble learning provides robustness against overfitting.

The results confirm that combining structured pre-processing, visualization-driven feature reduction, and ensemble classification offers an optimal balance between detection effectiveness and computational efficiency for cloud intrusion detection systems.

V. CONCLUSION

This research presented an optimized cloud-based intrusion detection framework that integrates structured preprocessing, visualization-driven feature engineering, and a Random Forest ensemble classifier to enhance anomaly detection performance while minimizing computational overhead. By reducing the feature space prior to classification, the proposed methodology effectively lowered model complexity, decreased execution time, and mitigated overfitting without compromising detection reliability. The ensemble voting mechanism of Random Forest improved classification stability and reduced variance compared to single-tree or traditional machine learning approaches. Experimental evaluation demonstrated high accuracy, precision, and recall, confirming the model's ability to correctly distinguish between normal and malicious traffic in dynamic cloud environments. Furthermore, computational complexity analysis showed significant improvements in scalability due to dimensionality reduction, making the system suitable for real-time deployment in resource-constrained infrastructures. The balanced trade-off achieved between detection effectiveness and efficiency validates the robustness of the proposed approach and highlights its practicality for securing modern cloud ecosystems against evolving cyber threats. Future work will focus on integrating adaptive deep ensemble learning with real-time streaming analytics to further improve detection of zero-day and highly sophisticated attacks.

VI. REFERENCES

[1] M. Ali et al., “Security in cloud computing,” Information Sciences, 2015.
[2] A. Singh and K. Chatterjee, “Cloud security issues and challenges,” Journal of Network and Computer Applications, 2017.
[3] Q. Zhang et al., “Cloud computing: State-of-the-art and research challenges,” J. Internet Serv. Appl., 2010.
[4] Z. Chiba et al., “Hybrid network intrusion detection framework,” Procedia Computer Science, 2016.
[5] A. Khraisat et al., “Survey of intrusion detection systems,” Cybersecurity, 2019.
[6] A. L. Buczak and E. Guven, “Data mining and ML for cybersecurity IDS,” IEEE Communications Surveys & Tutorials, 2016.
[7] N. Chand et al., “Comparative analysis of SVM for intrusion detection,” 2016.
[8] N. M. Abdulkareem and A. M. Abdulazeez, “Random Forest classification review,” 2021.
[9] L. Zhou et al., “Cyber-attack classification via DNN,” 2018.
[10] F. Jiang et al., “Deep learning multi-channel intrusion detection,” IEEE TSC, 2018.
[11] S. Potluri and C. Diedrich, “Accelerated deep neural networks for IDS,” 2016.
[12] L. Breiman, “Random Forests,” Machine Learning, 2001.
[13] I. Reis et al., “Probabilistic Random Forest,” 2018.
[14] A. Devarakonda et al., “Comparative study using NSL-KDD,” 2022.
[15] M. Shafiq et al., “Malicious Bot-IoT traffic detection,” IEEE IoT Journal, 2021.
[16] M. Shafiq et al., “Bot-IoT attack identification,” Future Generation Computer Systems, 2020.
[17] M. Hossin and M. N. Sulaiman, “Evaluation metrics for data classification,” 2015.
[18] T. A. Tang et al., “Deep learning for network intrusion detection,” 2016.
[19] A. Mishra et al., “ML-based DDoS detection in cloud,” 2021.
[20] B. A. Tama and K. H. Rhee, “Hybrid feature selection and tree ensemble IDS,” 2017.