

# Improving Accuracy of Intrusion Detection System using Stacked Ensemble Model and Voting Classifier

Mahbub Alam

*Computer Science and Engineering  
United International University  
Dhaka, Bangladesh  
malam212082@bscse.uiu.ac.bd*

Tamim Abdullah

*Computer Science and Engineering  
United International University  
Dhaka, Bangladesh  
tabdullah211072@bscse.uiu.ac.bd*

Mahmudur Rahman Tushar

*Computer Science and Engineering  
United International University  
Dhaka, Bangladesh  
mtushar202080@bscse.uiu.ac.bd*

Md.Shamsul Huda

*Computer Science and Engineering  
United International University  
Dhaka, Bangladesh  
mhuda211120@bscse.uiu.ac.bd*

Md.Sabit Hasan

*Computer Science and Engineering  
United International University  
Dhaka, Bangladesh  
mhasan191204@bscse.uiu.ac.bd*

**Abstract**—In response to the dynamic nature of cyber threats, establishing robust intrusion detection systems is imperative for preserving network integrity. This proposal outlines our intention to address the challenges of network intrusion detection through the strategic application of machine learning methodologies. We plan to utilize a diverse set of classifiers, including Naive Bayes, Logistic Regression, Random Forest, and Voting Classifier, leveraging the extensive CICIDS2017 dataset. Our research will encompass a comprehensive evaluation of these classifiers, focusing on their effectiveness in detecting network intrusions. While preliminary investigations show promising outcomes, our primary objective is to elucidate the strengths and weaknesses of each algorithm within the context of intrusion detection. Through this study, we aim to highlight the pivotal role of machine learning in enhancing network security and fostering a deeper understanding of intrusion detection mechanisms. The insights gleaned from our research will lay the groundwork for the development of more resilient and adaptive intrusion detection systems, equipped to combat the evolving cyber threat landscape.

**Index Terms**—network intrusion detection, machine learning, network security

## I. INTRODUCTION

A network intrusion detection system is employed to identify unwanted or malicious traffic within a network, playing a crucial role in maintaining the security of digital assets. As the prevalence and sophistication of cyber-attacks continue to rise, the need for accurate and efficient intrusion detection becomes paramount. In recent times, the landscape of intrusion detection has witnessed a shift towards advanced techniques, particularly machine learning (ML) algorithms. Thaseen et al. (2020) explored the application of various ML algorithms, achieving notable accuracy in detecting network intrusions. Ghani et al. (2023) introduced an effective Feedforward Neural Network (FFNN) classifier for network traffic anomaly de-

tection, demonstrating proficiency on diverse dataset. Hidayat et al. (2019) proposed a hybrid feature selection technique, optimizing accuracy and minimizing false positives and negatives. Injadat et al. (2020) presented a multi-stage optimized ML-based NIDS framework, surpassing recent experiments in intrusion detection accuracy. This study contributes to ongoing efforts to protect digital networks against rising threats by focusing on the accuracy and efficacy of intrusion detection systems. In a nutshell, here are our main contributions:

- Collection of a Latest Comprehensive Dataset for Network Intrusion Detection
- Comparative Analysis with Recent ML Techniques
- Enhancement of the Overall Detection Accuracy of the ML Model

This paper is structured as follows: Section II delves into the literature review and related works, Section III outlines our methodology, and Sections IV and V present the results, conclusion and future work respectively.

## II. LITERATURE REVIEW

With the expanding variety of cyber-attacks, the development of reliable and effective intrusion detection systems is essential for protecting sensitive network information. Numerous studies have been conducted to tackle these challenges and improve the accuracy of network intrusion detection. Thaseen et al. addressed various machine learning algorithms for detecting network intrusion more accurately. They used a dataset which is comprised by the Wireshark captured data over a network. The dataset contains a number of 1130 instances followed by a number of 24 attributes which are basically represents integer, decimal and nominal values. They evaluated various machine learning (ML) algorithms like Naive Bayes, Logistic Regression, Random Forest and Voting

Classifier on this dataset and achieved remarkable accuracy of 82.34% , 96.95%, 99.95%, 99.85% respectively for classifying encrypted packets, unencrypted packets, unencrypted malicious packets and encrypted malicious packets. Ghani et al. provides an effective technique to evaluate the classification performance of a deep learning-based Feed-forward Neural Network (FFNN) classifier. A small feature vector detects network traffic anomalies in the UNSWNB15 and NSL-KDD datasets. In this research, they used the feedforward neural network on the reduced feature set of the UNSW-NB15 dataset and achieved 91.29% accuracy, a 91.38% detection rate, and an 8.79% false positive rate. Using the same classifier on the reduced feature set of the NSL-KDD dataset They achieved 89.03% accuracy, a 95.65% detection rate, and a 17.59% false positive rate. Hidayat et al. proposed a hybrid feature selection technique composed of the Pearson correlation coefficient and random forest model for intrusion detection. They used various machine learning (ML) and deep learning (DL) techniques like the decision tree, AdaBoost, and Knearest neighbor, multilayer perceptron (MLP) and long short-term memory for training and testing the TON IoT dataset. They achieved an accuracy of 99.6% by the decision tree, 99.8% by AdaBoost, and 99% by KNN. Their study provides optimal accuracy with fewer false-positive and false-negative rates on the TON IoT dataset. Injadat et al. proposed a novel multi-stage optimized ML-based NIDS framework for network intrusion detection. To reduce the computational complexity while maintaining performance, they compared information gain and correlation-based feature selection techniques and investigated various ML hyper-parameter optimization techniques. They evaluated their proposed framework on the CICIDS 2017 and the UNSW-NB 2015 datasets. They achieved an accuracy of 99% for both datasets which outperformed recent experiments in this domain. Rizvi et al. [10] introduced a flexible and efficient intrusion detection systems using deep learning techniques. They evaluated their proposed 1D-DCNN model on two different datasets namely CIC-IDS2017 dataset which includes more than 80 characteristics and 15 classes and CSE-CIC-IDS2018 dataset which contains approximately 80 various types of features. Their model achieved 99.7% and 99.98% accuracy respectively on CICIDS2017 and CSE-CIC-IDS2018 datasets. In future, they will explore the effectiveness of the model comparing other state-of-the-art deep learning approaches in terms of performance and computing cost. Cao et al. presented the diverse array of models and approaches used in the realm of intrusion detection. The proposed model addresses data set imbalance and feature redundancy issues through the ADRDB and RFP algorithms. By combining CNN and GRU, it achieves comprehensive feature learning. Also, it incorporates an attention module to reduce overhead and enhance model performance. They evaluated their model on the NSL-KDD dataset, UNSW NB15 dataset and CIC-IDS2017 dataset and achieved an accuracy of 99.69%, 86.25% and 99.65%, respectively. However, they will focus on time cost and overall detection improvement in future. Yang et al.

conducted a comprehensive literature survey focused on

anomaly-based network intrusion detection, aiming to shed light on recent techniques and datasets in this field. With a meticulous approach, the authors scrutinized 119 of the most highly-cited papers in the realm of anomaly-based intrusion detection, exploring various facets of the technical landscape. Their investigation encompassed diverse aspects, including the application domains where these techniques are applied, data preprocessing methods, and the attackdetection techniques employed. Furthermore, they delved into the evaluation metrics commonly used to assess the effectiveness of these approaches, and even examined coauthor relationships and datasets utilized in this domain. The ultimate goal of this study was to contribute to the advancement of research in this field by pinpointing unresolved challenges and unexplored research topics. In addition, the paper outlines several prospective and highimpact research directions that hold promise for the future of network intrusion detection. Within the literature review section, the authors provide a comprehensive analysis of their findings and offer valuable recommendations, consolidating their contribution to the evolving landscape of network intrusion detection research. Alavizadeh et al.

introduced a a Deep Q-learning based (DQL) reinforcement learning model for network intrusion detection. Their proposed method combines a Q-learning based reinforcement learning with a deep feed forward neural network. They evaluated their method on the NSL-KDD dataset and achieved high accuracy for classifying different network intrusion attack types. They will deploy their proposed method on a realistic cloud-based environment in future. Apruzzese et al. promoted the idea for crossevaluation of ML-NIDS utilizing existing labelled data from multiple networks. They used their framework XeNIDS for accurate cross-evaluations based on Network Flows. Using XeNIDS on six well-known datasets, they highlighted the concealed potential, and the risk of ML-NIDS crossevaluations. In the future, they will demonstrate the contexts that may be analyzed by combining diverse network dataset. However, a machine learning (ML) based classifier can accurately detect network intrusion while handling the large amount of data.

### III. METHODOLOGY

#### A. Data Collection

1) *Dataset Description:* The research utilized the CIC-IDS-2017 dataset, consisting of multiple CSV files representing various cyberattack scenarios, including DDoS attacks, port scans, web attacks, and infiltration, recorded during different days and hours.

2) *Data Preparation:* The individual files inside CIC-IDS-2017 dataset are merged into a unified Data Frame. During this process, missing and infinite values were carefully addressed to ensure the integrity of the dataset.

#### B. Exploratory Data Analysis (EDA)

1) *Feature Categorization:* Features were categorized as numerical or categorical, with emphasis on detecting the label column for binary classification.

2) *Label Mapping*: To facilitate binary classification, labels were mapped, designating “BENIGN” as the normal class and categorizing all other instances as “INTRUSION”.

3) *Data Inspection*: During exploratory data analysis, we thoroughly examined the distribution of categorical features and handled missing values in numerical features.

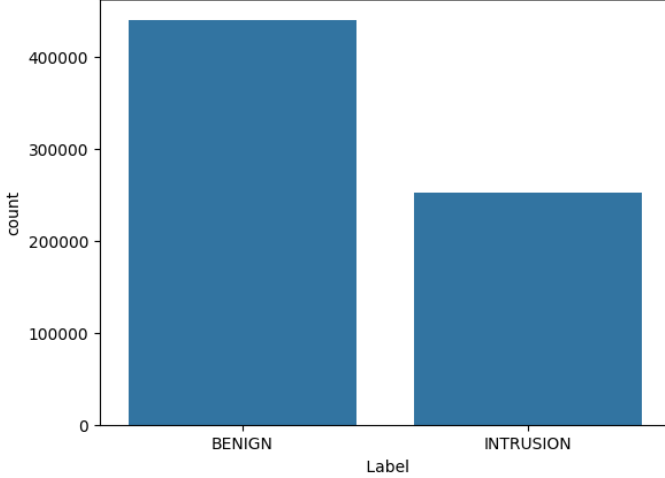


Fig. 1. Label Mapping from CIC\_IDS\_2017 dataset

### C. Data Transformation

1) *Feature Scaling*: Numerical features were normalized using Min-Max scaling to ensure consistent ranges across the dataset. The Min-Max Normalization formula is given by:

$$X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Here,  $X$  represents the original data point,  $X_{\text{normalized}}$  is the scaled value,  $X_{\min}$  is the minimum value in the dataset, and  $X_{\max}$  is the maximum value in the dataset.

2) *One-Hot Encoding*: Categorical labels were converted into numerical format using one-hot encoding, enabling machine learning models to process these features effectively.

### D. Data Splitting

A crucial step was to split the dataset into training and testing sets. This allowed for unbiased model evaluation and testing.

### E. Model Implementation and Training

1) *Decision Tree Classifier*: A Decision Tree Classifier is a model that resembles a tree and makes decisions by dividing the data into subsets based on features. It continues to split the data recursively to create a tree structure, where each leaf node represents either a class or a regression value.

2) *Naive Bayes Classifier*: The Naive Bayes Classifier is a statistical model that utilizes Bayes’ theorem. It makes an assumption that the features are independent of each other, given the class label. Although this assumption seems naive, the model is computationally efficient and performs well in text classification and various other tasks.

3) *Logistic Regression*: Logistic Regression is a model used for binary classification, which predicts the probability of an instance belonging to a particular class. It utilizes the logistic function to compress the linear prediction between 0 and 1, making it ideal for probability estimation.

4) *Random Forest Classifier*: A Random Forest Classifier is a machine learning algorithm that creates several decision trees during training and provides the mode of the classes for classification or the mean prediction for regression. This ensemble method is useful for enhancing accuracy and controlling overfitting.

5) *Voting Classifier*: Voting Classifier combines predictions from multiple base classifiers through majority voting. By leveraging the collective intelligence of diverse models, it enhances classification accuracy and robustness. It’s particularly useful when individual models excel in different areas or exhibit complementary strengths.

6) *Stacking*: Stacking is a meta-ensemble learning technique that combines predictions from multiple base models using a higher-level model. By learning how to optimally combine diverse model outputs, stacking enhances predictive performance and robustness. It’s suitable for complex classification tasks where individual models may struggle to capture all aspects of the data.

### F. Model Evaluation and Interpretation

1) *Confusion Matrix Analysis*: Confusion matrices were used to display and interpret true positives, true negatives, false positives, and false negatives of the models.

2) *Feature Importance*: The significance of different predictors was visualized to interpret the importance of features, particularly for tree-based models.

### G. Performance Metrics and Comparison

1) *Metric Evaluation*: Performance metrics, such as accuracy, precision, recall, and F1 score, were systematically evaluated and compared across different models.

2) *ROC and Precision-Recall Curves*: Receiver Operating Characteristic (ROC) and Precision-Recall curves were plotted, providing a nuanced understanding of model performance at varying thresholds.

### H. Proposed Learning Model

1) *Base Classifiers Selection*: : We begin by selecting a diverse set of base classifiers, denoted as  $h_1, h_2, \dots, h_n$  each capturing unique aspects of network traffic behavior. These classifiers are chosen based on their complementary strengths and ability to discriminate between normal and malicious network activities.

2) *Training of Base Classifiers*: : Each base classifier  $h_i$  is trained independently on the feature vectors extracted from network packet data. Let  $X$  represent the feature matrix, where each row corresponds to a network packet and each column represents a specific feature. During training, the classifiers learn to predict the binary label  $y$  indicating whether a packet is benign ( $y = 0$ ) or malicious ( $y = 1$ ).

3) *Meta-Classifiers Construction*: : The predictions of the base classifiers on the training dataset are aggregated to form the input for the meta-classifier. We employ a soft voting mechanism, where the final prediction  $\hat{y}$  is determined by combining the weighted outputs of the base classifiers:

$$\hat{y} = \underset{y}{\operatorname{argmax}}(\sum_{i=1}^n w_i * h_i(X))$$

where  $w_i$  represents the weight assigned to each base classifier. These weights can be determined based on their performance metrics or through ensemble optimization techniques such as meta-learning or genetic algorithms.

4) *Model Optimization and Refinement*: : The stacked ensemble model undergoes iterative optimization and refinement processes to improve its performance further. This may involve hyperparameter tuning, feature selection, or ensemble pruning techniques to enhance the model's generalization and robustness.

#### IV. RESULT ANALYSIS

##### A. Analysis of our proposed model

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

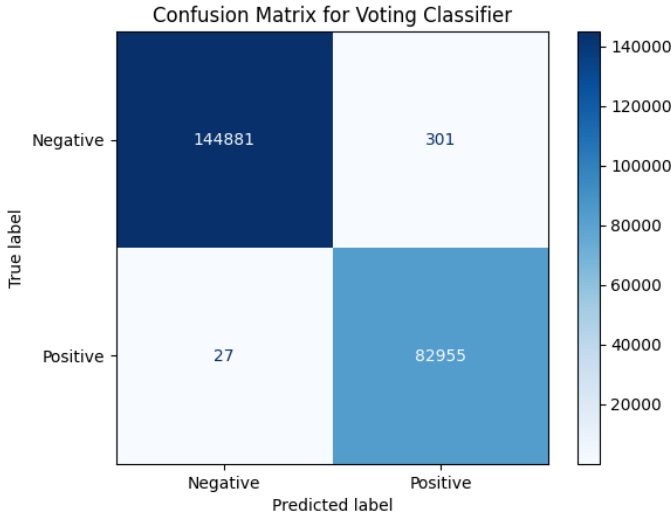


Fig. 2. voting classifier

##### B. Comparison of our Proposed Model against Other Machine Learning Algorithms

The Naive Bayes Classifier shows relatively lower performance metrics compared to the Stacking Ensemble Learning. While it demonstrates moderate accuracy, its F1 score, precision, and recall are notably lower, indicating limitations in effectively capturing the intricacies of the dataset.

Logistic Regression performs moderate in accuracy and F1 score, but its precision and recall may be lower than Decision Tree Classifier, potentially affecting its ability to classify network intrusions correctly. The Random Forest and Voting

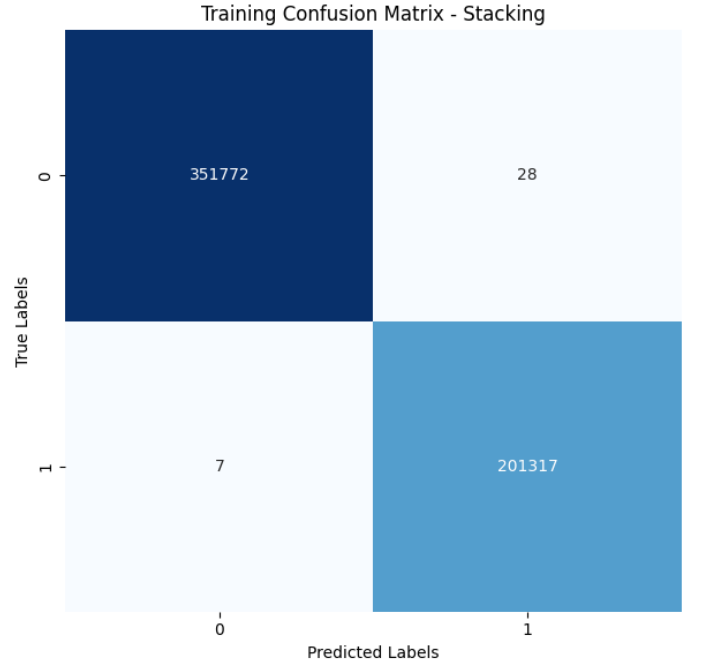


Fig. 3. stacked learning on training

Our stacked ensemble model demonstrated superior performance across various evaluation metrics compared to individual base classifiers. The model achieved an accuracy of 99.99%, precision of 99.94%, recall of 99.97%, and F1 score of 99.95% on the test dataset. These metrics highlight the model's ability to accurately differentiate between normal and malicious network activities.

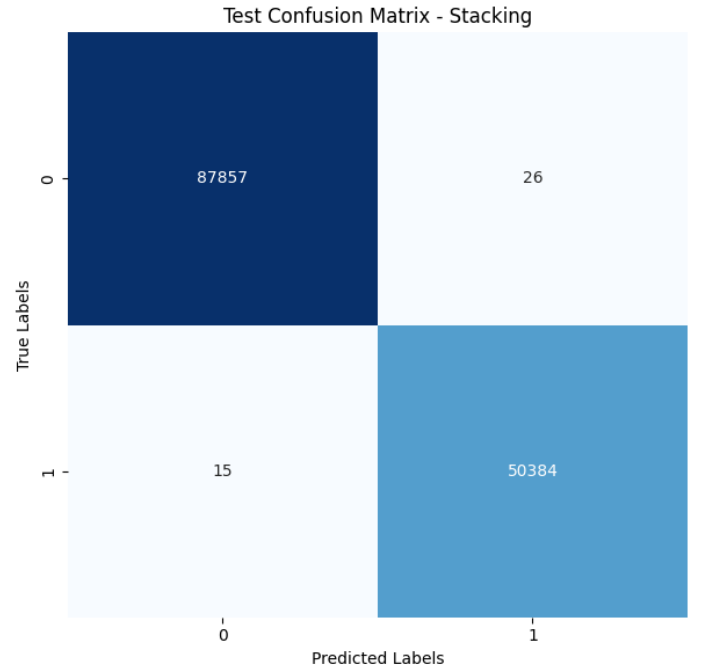


Fig. 4. stacked learning on testing

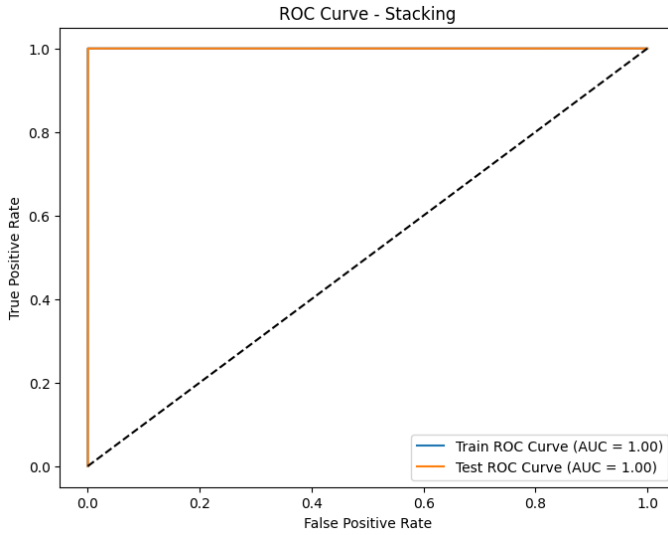


Fig. 5. ROC curve of stacking

ROC analysis revealed the model's robustness in distinguishing between benign and malicious traffic across different operating points. The area under the ROC curve (AUC) was measured at 1.00, indicating high discriminatory power and minimal false positive rates. This suggests that our model effectively balances sensitivity and specificity in detecting network intrusions.

TABLE I  
COMPARISON AMONG DIFFERENT ML MODELS

| Model                  | Train Acc. | Test Acc. | F1 Score | Precision | Recall |
|------------------------|------------|-----------|----------|-----------|--------|
| Naive Bayes Classifier | 82.15%     | 82.34%    | 78.82%   | 69.90%    | 90.35% |
| Random Forest          | 99.96%     | 99.95%    | 99.93%   | 99.90%    | 99.96% |
| Logistic Regression    | 96.93%     | 96.95%    | 95.82%   | 95.60%    | 96.03% |
| Voting Classifier      | 99.89%     | 99.85%    | 99.80%   | 99.63%    | 99.96% |
| Proposed Model         | 99.99%     | 99.97%    | 99.95%   | 99.94%    | 99.97% |

Classifier performs admirably, showcasing high accuracy and F1 score, almost comparable to the Decision Tree Classifier. Its precision and recall rates are also impressive, suggesting robustness in identifying network intrusions.

## CONCLUSION AND FUTURE WORK

This research underscores the effectiveness of using stacking method of ensemble learning with voting classifier as the base model to improve the intrusion detection system. The model exhibits outstanding performance, achieving a high F1 Score and accuracy. The detailed examination further sheds

light on the strengths and weaknesses of Naive Bayes, Random Forest Classifier, Decision Tree classifiers, KNN Classifier and Logistic Regression. The delicate balance between accurately identifying intrusions and minimizing false positives is elucidated through precision and recall measurements. We propose exploring ensemble approaches to enhance the effectiveness of machine learning models in network security. Our work contributes to the development of efficient intrusion detection systems, crucial for staying ahead of emerging threats and ensuring network security.

## REFERENCES

- [1] Humera Ghani, Bal Virdee, and Shahram Salekzamankhani. A deep learning approach for network intrusion detection using a small features vector. *Journal of Cybersecurity and Privacy*, 3(3):451–463, 2023.
- [2] Giovanni Apruzzese, Luca Pajola, and Mauro Conti. The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Transactions on Network and Service Management*, 19(4):5152–5169, 2022.
- [3] Bo Cao, Chenghai Li, Yafei Song, Yueyi Qin, and Chen Chen. Network intrusion detection model based on cnn and gru. *Applied Sciences*, 12(9):4184, 2022.
- [4] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01):20–28, 2021.
- [5] Abhik Das. Logistic regression. In *Encyclopedia of Quality of Life and Well-Being Research*, pages 1–2. Springer, 2021.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa. "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [7] Imran Hidayat, Muhammad Zulfiqar Ali, and Arshad Arshad. Machine learning-based intrusion detection system: An experimental comparison. *Journal of Computational and Cognitive Engineering*, 2(2):88–97, 2023.
- [8] MohammadNoor Injadat, Abdallah Moubayed, Ali Bou Nassif, and Abdallah Shami. Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Transactions on Network and Service Management*, 18(2):1803–1816, 2020.
- [9] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. A review on random forest: An ensemble classifier. In *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, pages 758–763. Springer, 2019.
- [10] Syed Rizvi, Mark Scanlon, Jimmy McGibney, and John Sheppard. Deep learning based network intrusion detection system for resource-constrained environments. In *International Conference on Digital Forensics and Cyber Crime*, pages 355–367. Springer, 2022.
- [11] Mucahid Mustafa Saritas and Ali Yasar. Performance analysis of ann and naive bayes classification algorithm for data classification. *International journal of intelligent systems and applications in engineering*, 7(2):88–91, 2019.
- [12] Amin Shokrzade, Mohsen Ramezani, Fardin Akhlaghian Tab, and Mahmud Abdulla Mohammad. A novel extreme learning machine based knn classification method for dealing with big data. *Expert Systems with Applications*, 183:115293, 2021.
- [13] I Sumaiya Thaseen, B Poorva, and P Sai Ushasree. Network intrusion detection using machine learning techniques. In *2020 International conference on emerging trends in information technology and engineering (IC-ETITE)*, pages 1–7. IEEE, 2020.
- [14] Zhen Yang, Xiaodong Liu, Tong Li, Di Wu, Jinjiang Wang, Yunwei Zhao, and Han Han. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers and Security*, 116:102675, 2022.