Enhancing Network Security: A Stacking Ensemble Approach for Intrusion Detection Systems

V.Mounika[1],

[1]Asst Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.vmounika@kluniversity.in

Dr.N.Raghavendra Sai[2]

[2]Asst Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India..

nallagatlaraghavendra@kluniversity.in

Abstract:

Modern cyber security relies heavily on intrusion detection in network traffic to quickly identify and mitigate security threats. Conventional intrusion detection systems frequently depend on lone, stand- alone models that could find it difficult to change with the ways that network attacks are evolving. In order to overcome this difficulty, we provide an ensemble-based strategy that improves threat detection effectiveness and accuracy by utilizing the strength of several intrusion detection models. Increasing security measures against data breaches and network invasions is essential due to the ever-growing usage of the Internet and networks. As intrusions are frequently hidden within of valid network packets, firewalls have a difficult time identifying and stopping them. Furthermore, the majority of network monitoring systems and algorithms find it increasingly difficult to handle the sheer volume of network traffic. Various intrusion detection strategies have been proposed in response to these issues, with machine learning techniques emerging as a possible route for handling these situations. In this paper, an intrusion detection system (IDS) that makes use of stacking ensemble learning is presented. The three fundamental machine learning models that make up the core ensemble are k-nearest-neighbours, Decision Tree, and Random Forest. In order to improve classification performance, the suggested system combines a total of seven machine learning algorithms with pre-processing methods. By merging the outputs of the underlying models with a meta-model embodied by the Logistic Regression algorithm, the stacking ensemble technique improves performance. The UNSW-NB15 dataset is used to assess the efficacy of the

IDS. The suggested IDS obtains an astounding 96.16% accuracy rate in the training phase and an even greater 97.95% accuracy rate in the testing phase. Impressive precision scores as well—97.78% during training and 98.40% during testing—are obtained. The system's capacity to identify and reduce network intrusions is demonstrated by these results, which show notable enhancements across a range of measuring criteria.

Keyword: Intrusion detection system (IDS); machine learning techniques s; stacking ensemble; random forest; decision tree; k-nearest-neighbour

1.Introduction

The Internet's explosive growth and the widespread use of networks in the modern digital age have completely changed how we interact, collaborate, and transact business. Unprecedented levels of ease and connectedness have been brought about by this technological revolution, but it has also created a formidable obstacle in the form of the growing threat of data breaches and network attacks. These intrusions, carried out via network packets, pose a continuous and changing threat to the confidentiality and safety of our digital infrastructure.

In order to counter these threats, one of the main challenges is the misleading resemblance between regular network traffic and intrusion efforts. The large stream of normal data that intruders frequently use to conceal their malicious activity makes it very difficult for firewalls and other traditional security measures to tell friend from foe.

Furthermore, classic network monitoring systems and rule-based algorithms face major problems from the sheer volume and complexity of network traffic in today's interconnected world. It's getting harder to find anomalies and possible dangers in this deluge of data, which calls for more advanced and flexible solutions.

The discipline of intrusion detection has seen a growth of creative ideas and methodology in response to this urgent requirement. Among these, machine learning approaches have attracted a lot of interest due to their potential to improve intrusion detection systems' capabilities. Machine learning has potential in enhancing the precision and effectiveness of detecting and addressing security events by utilizing algorithms and data-driven analyses.

Network security has become critical in an era where communication technologies and the

Internet are developing quickly. Businesses all over the world are forced to make large investments in protecting their critical operations and sensitive data. They use a variety of security measures, like as intrusion detection systems (IDS), firewalls, and antivirus software, to accomplish this, all with the goal of protecting the integrity and confidentiality of their networks and related assets. Among these, intrusion detection systems (IDS) are essential because they actively detect anomalous network activity and rapidly alert network administrators to it [1].

In the last ten years, network security has seen a significant rise in the use of machine learning techniques. These methods have become quite popular because of their exceptional capacity to extract relevant features from network data and then apply learnt patterns to produce precise classifications [2]. Deep Learning (DL)-based intrusion detection systems (IDS) are particularly noteworthy because of their deep neural network topologies, which enable them to independently extract complex features from unprocessed data and provide a powerful defense against new threats [3].

Ensemble learning is another potent machine learning paradigm. To solve certain computational intelligence problems, several models—such as classifiers or domain experts—are built and carefully integrated through ensemble learning. Ensemble learning is used in many different fields, and its major goals are to reduce the risk resulting from inadvertently choosing a weak model and to improve model performance (e.g., classification, prediction, function approximation).

Within this framework, the study explores the mutual enhancement of these two powerful tools: ensemble learning and machine learning, specifically concentrating on Deep Learning. Our goal is to investigate how their combination can improve intrusion detection systems' efficacy. We look into how combining these approaches can result in more precise and adaptable systems that can protect against the always changing array of network threats.

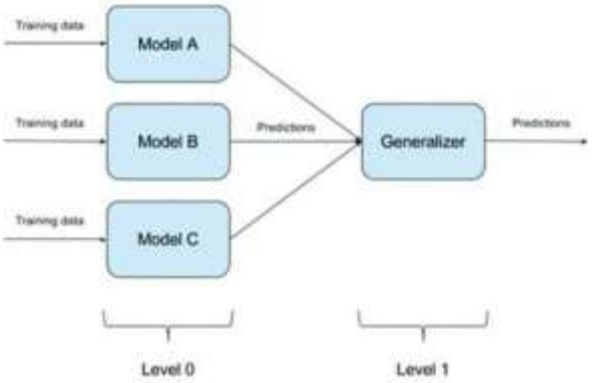


Fig 1: Ensemble classifiers basic architecture

[6]. Being a watchful defender against illegal activity is the detection system's job within an intrusion detection system (IDS).

As a result, an IDS acts as a watchful security instrument, constantly keeping an eye on host and network activity. Its main goal is to identify any indications of questionable activity that defies set security guidelines and compromises the network's availability, confidentiality, and integrity. An IDS is usually attached to a network adapter that is set up using port mirroring technology in order for it to properly perform its function, as shown in Fig. 1. The intrusion detection system (IDS) instantly notifies the network administrator when it detects harmful activity.

In IDS, Deep Learning and Machine Learning Researchers and practitioners have investigated the integration of machine learning (ML) and deep learning (DL) technologies within intrusion detection systems (IDS) in response to the always changing threat landscape. To extract valuable insights from large datasets, machine learning and deep learning are both essential. Network security has embraced these technologies more quickly in the last ten years due to the widespread use of powerful Graphics Processing Units (GPUs) [7, 8]. When it comes to analyzing and predicting network traffic, ML and DL approaches excel. Interestingly, although DL-based IDS uses its deep neural network architecture to automatically learn complex features directly from raw data, ML-based IDS frequently depends on engineered features to extract meaningful information from network traffic data [2][3].

Modules of Function in IDS Architecture

As shown in Fig. 2, the architecture of an IDS is usually made up of four main functional modules: Event- Boxes, Database-Boxes, Analysis-Boxes, and Response-Boxes. As sensors, Event-Boxes keep an eye on the system all the time, gathering pertinent data for further examination. The Database-Boxes serve as the repository for the gathered data, guaranteeing that the data is accessible for processing. The fundamental processing module is represented by the Analysis-Boxes, wherein dangerous conduct is detected through a close examination of the gathered events. The most important stage is reacting quickly to any malicious activity that is identified; the Response-Boxes are responsible for this. This design is shown visually in Fig. 2, which is an example of the IDS framework [1].

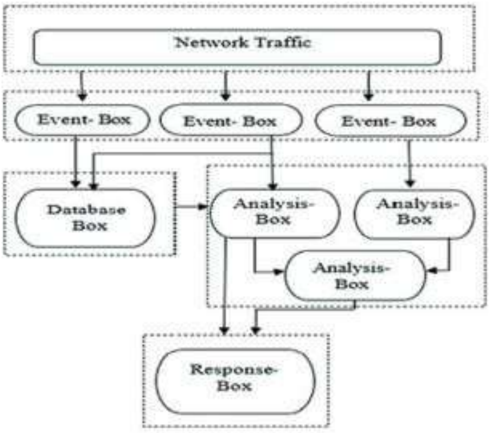


Fig 2: Common intrusion detection architecture for IDS

IDSs that are Host-Based (HIDS) and Network-Based (NIDS)

An IDS's Event-Boxes can be classified into two main types based on where the information comes from: Host-Based IDS (HIDS) and Network-Based IDS (NIDS). Monitoring system calls and process identifiers is

cover both packet- and flow-based variables, providing a thorough understanding of network traffic behavior.

Four separate groups can be formed from these features:

1. Content-Based characteristics: These characteristics include details on the contents of network packets, giving information about the data that is carried in the payload.

2. Fundamental Features: These features serve as the cornerstone for analysis by capturing the crucial qualities and traits of network traffic.

3. Flow-Based Features: These features are centered on tracking and keeping an eye on data flows that occur within the network, giving important insights into data transfer

trends.

Time-Based Features: By emphasizing temporal elements, time-based features allow network activity patterns to be analyzed over time.

3. Data Acquisition

To assess and evaluate our proposed approach, we utilized the UNSW-NB15 dataset sourced from ACCS, recognized as a contemporary benchmark dataset for Network Intrusion Detection Systems (NIDS). This dataset comprises a substantial 2.5 million records, encompassing 45 distinctive features. To streamline our analysis, we performed some modifications on the original UNSW-NB15 dataset, reducing the feature set from 45 to 43. These features encompass both flow-based and packet-based attributes.

The features within the dataset can be categorized into four distinct groups, namely content, fundamental, flow, and time-based features. These subsets of features offer valuable insights into network traffic and behavior patterns.

Regarding the data volume, we carefully selected a subset of 257,673 data instances from the UNSW- NB15 dataset for our analysis. This subset was then further partitioned into two essential segments: training data instances, consisting of 175,341 records, and testing data instances, which include 82,332 records.

This streamlined dataset configuration allowed us to efficiently train and test our Intrusion Detection System (IDS) while preserving the most pertinent characteristics of the UNSW-NB15 dataset.

4. Outline of the Suggested System for Ensemble Learning

Ensemble learning is a popular machine learning technique that combines the best features of individual classifiers to produce a single classification model with improved overall classification and prediction performance. When compared to single, standalone classifiers, ensemble learning has demonstrated consistently better performance in the particular setting of intrusion detection systems (IDS). Developing a strong network intrusion detection system by utilizing ensemble classifiers' enormous potential is the main goal of this research.

To apply our suggested technique, datasets that are mainly intended for training must be carefully preprocessed. As the basis for our work, we have employed the UNSW-NB15 dataset, which is a large collection of 45 unique attributes. These features consist of four nominal qualities and forty-one numerical attributes, the latter of which were converted to numerical values in order to aid in the process of analysis and model building.

model's convergence and overall performance while also lessening the effect of feature

$$y = \frac{x - min}{\max - min}$$

magnitude variations.

5. Architecture of the Proposed Approach

This study's primary goal is to identify network intrusions with accuracy. In order to accomplish this, our suggested method is divided into two basic levels, each of which is essential to the intrusion detection procedure. The Base-Learner layer and the Combining-Module layer are two of these layers that are shown in Figure 6.

Layer of Base-Learner: The Base-Learner layer, which is the initial layer, forms the basis of our intrusion detection system. We carefully choose three base binary classifiers in this layer, each of which is intended to detect intrusion patterns:

1. Random Forest Classifier (R.F.): Capable of handling complicated issues, this classifier makes use of an ensemble of decision trees. Prediction accuracy is increased by using numerous decision trees on various dataset subsets and averaging their outcomes.

2. Decision Tree Classifier (D.T.): This algorithm is highly effective at extracting useful data from large datasets. It makes value predictions using a test dataset to gauge its accuracy and a training dataset to build a decision tree.

3. K-Nearest Neighbors (K-NN) Classifier: Based on the available data, K-NN determines the probability that a given data point belongs to the closest group.

The Combining-Module layer receives the combined output of these base classifiers, which enhances the system's overall intrusion detection capabilities.

Layer of Combining Modules: As an aggregator, the Combining-Module layer balances the output produced by each of the separate basic classifiers. By significantly improving prediction accuracy, this aggregation technique makes the system more robust and trustworthy in detecting network intrusions.

Stack generalization is then used to direct the first layer's ensemble output into a meta-classifier. In this instance, the meta-classifier uses logistic regression to forecast the likelihood of a particular class or event, like pass/fail or win/lose.

Group Learning Subjects: Three main methods of ensemble learning can be broadly classified as follows: bagging, boosting, and stacking. Of them, bagging is the most widely used technique to forecast test results. In order to improve performance, models are extensively trained on misclassified data during the boosting process. In this work, stacking—also referred to as stacked generalization—is the preferred ensemble technique. It is mainly used to improve classification performance by combining several classifiers, as shown in Figure 7.

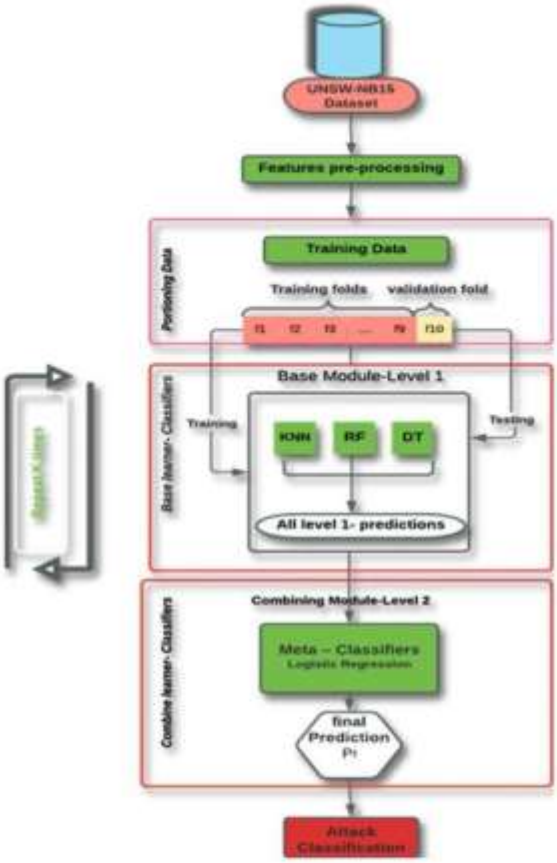


Figure 7: IDS based on stacking

Stacking Ensemble Learning: Unlike bagging and boosting, stacking involves two separate levels: level 0 for the base learner and level 1 for the meta-learner. Heterogeneous classification models learn from the training set at the first level, and the outputs of these models provide a fresh dataset for the stacking learner. This dataset uses the level 0 prediction as an attribute and associates each instance with the right class to be predicted.

The final result is then generated by the meta-learner using this newly created dataset, as seen in Figure 8.

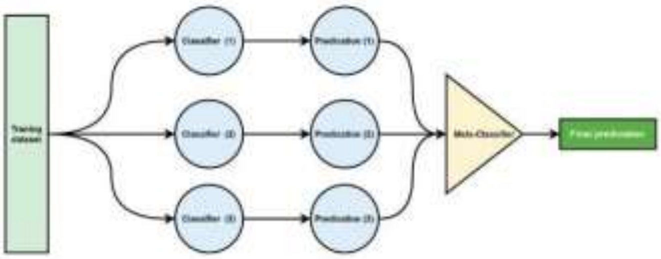


Figure 8: Stacking Classifiers

In summary, our proposed approach combines the strengths of multiple classifiers through a sophisticated ensemble learning framework. This two-layered architecture enhances the accuracy and reliability of intrusion detection by leveraging the diverse capabilities of the selected base classifiers and optimizing their outputs through the stack generalization process. Ultimately, this approach aims to provide a robust and effective solution for the detection of network intrusions.

6. Experimental Results and Analysis

We offer a thorough presentation and analysis of the experimental results from our suggested Intrusion Detection System (IDS), which is based on machine learning techniques and approaches, in this part. Using the UNSW-NB15 dataset, a well-known benchmark for network intrusion detection, we deployed seven different algorithms to the suggested IDS system in order to thoroughly evaluate its efficacy. A range of assessment criteria and performance metrics were used to evaluate the operation of our IDS system. These measurements are crucial reference points for determining how well the system detects network breaches.

The following key assessment and performance metrics were utilized to evaluate the performance of our proposed IDS:

1. **Accuracy:** Accuracy is the ratio of correctly classified instances to total instances, which indicates the overall correctness of the IDS. Better accuracy is indicated by higher numbers, which range from 0 to 1.

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

2. **Precision:** Precision assesses the system's accuracy in detecting intrusions among the occurrences it classifies as positive. It represents the percentage of real

positives out of all cases that are classified as positive.

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

3. **Recall:** Also known as sensitivity or true positive rate, recall evaluates the ability of the intrusion detection system to find all pertinent incursions. It measures the percentage of real positives among all real positive cases that have been found.

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

4. **AUC (Area Under the Receiver Operating Characteristic Curve):** The metric known as AUC, or Area Under the Receiver Operating Characteristic Curve, evaluates the overall effectiveness of the intrusion detection system by taking into account the trade-off between the true positive rate and the false positive rate. It offers a thorough evaluation of the discriminating power of the system.
5. **F1-Measure:** The harmonic mean of recall and precision is the F1-Measure. It provides a fair assessment of the IDS's effectiveness, especially in the case of unbalanced datasets.

$$F1 - Measure = 2 \times \frac{Precision \times Rrcall}{Precision + Rrcall}$$

6. **Mean Squared Error (MSE):** The Mean Squared Error (MSE) is a metric that assesses the degree of accuracy in the IDS's predictions by measuring the average squared difference between the actual and anticipated values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\text{Yi} - \text{Yi}^{\wedge})^{2}$$

By employing these diverse assessment criteria, we aim to offer a thorough and well-rounded evaluation of the proposed IDS approach. These metrics collectively shed light on the system's accuracy, precision, recall, discrimination power, and ability to handle imbalanced datasets.

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

7. Performance Evaluation of ML Algorithms

To evaluate the effectiveness of the suggested approach, we thoroughly investigated seven different classifiers. A few of these classifiers are Random Forest (R.F.), Logistic Regression (L.R.), Decision Tree (D.T.), k-Nearest Neighbors (KNN), linear Support Vector Machine (SVM), SVM with Radial Basis Function (RBF) kernel, and Naïve Bayes (N.B.). This evaluation's main goal was to determine how accurate our suggested approach was, using the UNSW-NB15 dataset in particular and utilizing the entire feature space with 42 characteristics for binary classification.

We used a 10-fold cross-validation approach, more precisely Stratified 10-fold cross-validation, to guarantee a strong evaluation. The dataset is split into ten subsets for this process; nine of these subsets are used to train the classifiers, and the tenth subset is used for testing.

Table 3 presents a summary of the findings of this thorough evaluation that includes important performance measures. It includes the accuracy, precision, recall, F1-Score, AUC, and MSE values of the suggested ML algorithms on the training dataset. The examination yielded some noteworthy results, such as: Random Forest (R.F.) is the best performer with the greatest accuracy score of 96.12%; Linear SVM is the best performer with the highest precision score of 99.79%, while it has a lower recall score of 91.21%.

- Although Naïve Bayes (N.B.) has an impressive 93.41% recall score, its accuracy is marginally lower.

  At 96.38%, Decision Tree (D.T.) exhibits the highest recall value. In addition, D.T. has less variance in recall and accuracy than the N.B. model, which raises its F1 score.

- Random Forest (R.F.) performs exceptionally well, with the lowest Mean Squared Error (MSE) of 0.0388 and the highest F1 score of 97.18%.

Ensemble Learning for Enhanced Performance

Our intrusion detection system's performance was substantially improved by utilizing the stacking ensemble method. This method combines a number of base models (level-0 models) with a meta-model (level-1 model) that collects the predictions produced by the base models.

We used Random Forest (R.F.), Decision Tree (D.T.), and k-Nearest Neighbors (KNN) as the basis models in our ensemble setup, with Logistic Regression (L.R.) serving as the meta-model. This ensemble method produced an accuracy of 96.16%, which is a significant improvement. intrusion detection system's accuracy is further improved by using the stacking ensemble method, highlighting the value of ensemble learning in improving performance.

Method	Accuracy	Precision	Recall	F1-score	AUC	MSE
SVM(RBF)	93.60	99.63	91.69	95.49	95.36	0.0640
Random forest(RF)	96.12	97.98	96.38	97.18	95.96	0.0388
Logistic regression(LR)	93.40	99.13	91.83	95.34	94.79	0.0660
Linear SVM	93.31	99.79	91.21	95.31	95.33	0.0669
Naïve Bayes(NB)	81.38	78.16	93.41	85.10	79.44	0.1862
Decision tree(DT)	95.00	96.27	96.38	96.33	94.23	0.0500
KNN	93.76	95.90	94.98	95.44	93.02	0.0624
Stacking ensemble	96.16	97.78	96.62	97.20	95.88	0.0384

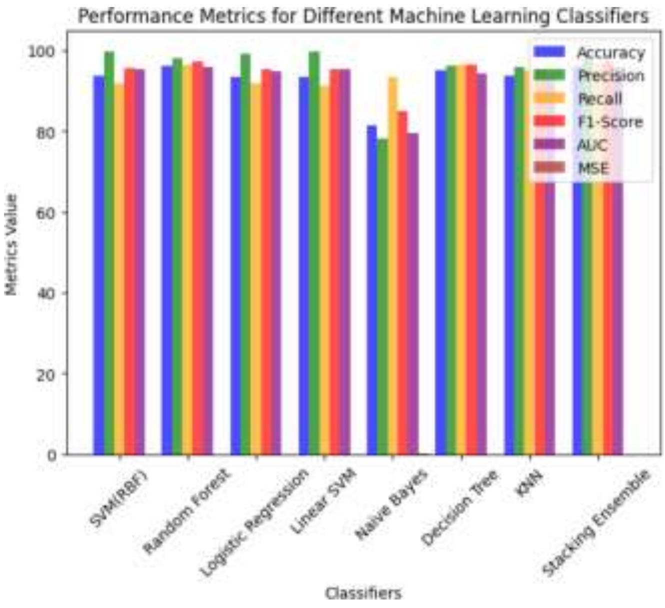
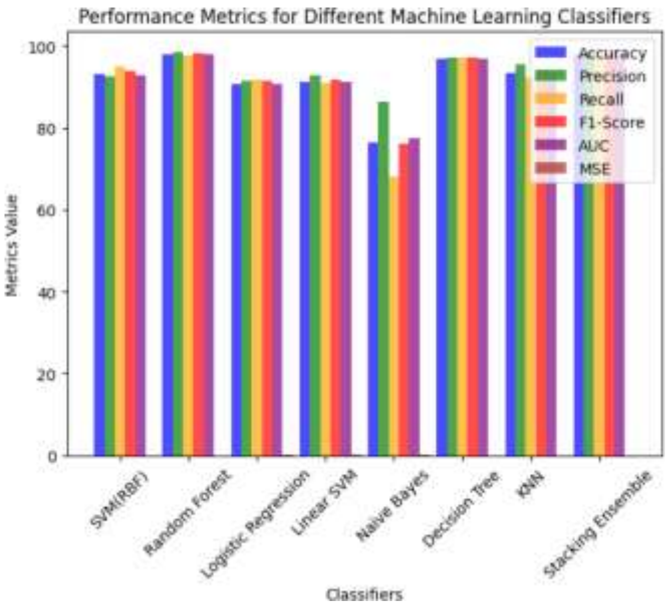


Table 4 shows the test dataset results for the suggested ML algorithms' accuracy, precision, recall, F1- Score, AUC, and MSE. R.F. achieves the greatest scores of 98.52%, 97.72%, 98.12%, 97.96%, and 98.54%, respectively, in accuracy, precision, recall, F1-score, and

AUC. D.T. comes in second with accuracy, precision, recall, and F1-score values of 96.74%, 96.98%, 97.11%, and 96.70%, respectively. KNN comes in third with accuracy, precision, recall, and F1-score values of 93.33%, 95.56%, 92.17%, and 93.46%, respectively. With an MSE error of 0.0206, R.F. has the best performance, followed by D.T. at 0.0326 and KNN at 0.0667. The accuracy achieved by the stacking ensemble is 97.95%.

A clearer explanation of the performance comparison of ML classifiers on the test dataset is provided in Fig. 9.

Method	Accuracy	Precision	Recall	F1-score	AUC	MSE
SVM(RBF)	93.10	92.65	95.01	93.81	92.89	0.0690
Random forest(RF)	97.94	98.52	97.72	98.12	97.96	0.0206
Logistic regression(LR)	90.70	91.54	91.58	91.56	90.61	0.0930
Linear SVM	91.04	92.70	90.89	91.78	91.06	0.0896
Naïve Bayes(NB)	76.43	86.28	68.02	76.07	77.38	0.2357
Decision tree(DT)	96.74	96.98	97.11	97.04	96.70	0.0326
KNN	93.33	95.56	92.17	93.84	93.46	0.0667
Stacking ensemble	97.95	98.40	97.87	98.13	97.96	0.0205



In conclusion, this study has successfully tackled the critical challenge of selecting the most suitable classifier for a specific classification task, particularly in the domain of intrusion detection within network data. Through rigorous experimentation and comparative analysis, various classifiers, including Multilayer Perceptron (MLP), Support Vector Machine (SVM), Decision Trees, and Naïve Bayes, were evaluated. The study's findings underscore the pivotal role of ensemble learning approaches, demonstrating their capacity to mitigate the risks associated with suboptimal classifier selection when compared to relying solely on a single classifier.

Conclusion and Future Directions

In conclusion, this study has successfully addressed the challenge of selecting the most suitable classifier for a specific classification task, such as intrusion detection in network data. The comparative analysis encompassed several classifiers, including Multilayer Perceptron (MLP), Support Vector Machine (SVM), Decision Trees, and Naïve Bayes. The key findings highlight the undeniable advantages of adopting an ensemble approach, where multiple classifiers are combined. This strategy significantly reduces the risk of making suboptimal choices compared to relying on a single classifier.One of the key contributions of this research lies in the application of the stacking ensemble technique. By leveraging base models such as Random Forest (R.F.), Decision Tree (D.T.), k-Nearest Neighbors (KNN), and a meta-model represented by Logistic Regression (L.R.), we achieved an impressive accuracy of 97.95% in the testing phase. This outcome showcases the potential of ensemble learning in enhancing the performance of Intrusion Detection Systems (IDS).

References

[1] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using gini impurity-based weighted random forest (GIWRF) feature selection technique," Cyberse- curity, vol. 5, no. 1, pp. 1–22, 2002. https://doi.org/10.1186/s42400-021-00103-8

[2] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald et al., "Deep learning applicationsand challenges in big data analytics," Journal of Big Data, vol. 2, no. 1, pp. 1–21, 2015. https://doi. org/10.1186/s40537-014-0007-7

[3] B. Dong and X. Wang, “Comparison deep learning method to traditional methods using for network intrusion detection,” in 8th IEEE Int. Conf. on Communication Software and Networks (ICCSN), Beijing, China,pp.581–585,2016.

[4] J. P. Anderson, “Computer security threat monitoring and surveillance,” Technical Report, James P. AndersonCompany,1980.

[5] H. Debar, M. Dacier and A. Wespi, “Towards a taxonomy of intrusion-detection systems,”ComputerNetworks,vol.31,no8,pp.805–822,1999. https://doi.org/10.1016/S1389-1286(98)00017-6

[6] S. Mukkamala, G. Janoski and A. Sung, “Intrusion detection using neural networks and support vector machines,” in 2002 Int. Joint Conf. on Neural Networks. IJCNN’02 (Cat. No. 02CH37290), Honolulu, HI, USA,vol.15,pp.1702–1707,2002.

[7] J. Lew, D. A. Shah, S. Pati, S. Cattell, M. Zhang et al., “Analyzing machine learning workloads using a detailed GPU simulator,” in IEEE Int. Symp. on Performance Analysis of Systems and Software (ISPASS), Madison,WI,USA,pp.151–152,2019.

[8] M. Abirami, U. Yash and S. Singh, “Building an ensemble learning based algorithm for improving intrusion detection system,” In: M. Abirami (Ed.), Artificial Intelligence and Evolutionary Computations in Engineering Systems, pp. 635–649, Singapore: Springer, 2000.

[9] A. Khraisat, I. Gondal, P. Vamplew and J. Kamruzzaman, “Survey of intrusion detection systems: Techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, pp. 1–22, 2019.https://doi.org/10.1186/ s42400-019-0038-7

[10] J. H. Liao, C. H. R. Lin, Y. C. Lin and K. Y. Tung, “Intrusion detection system: A comprehensive review,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013. https://doi.org/10.1016/j. jnca.2012.09.004

[11] H. Rajadurai and U. D. Gandhi, “A stacked ensemble learning model for intrusion detection in wirelessnetwork,”NeuralComputingandApplications,pp1–9,2000.

[12] X. Gao, C. Shan, C. Hu, Z. Niu and Z. Liu, “An adaptive ensemble machine learning model for intrusion detection,” IEEE Access, vol. 7, pp. 82512–82521, 2019. https://doi.org/10.1109/ACCESS.2019.2923640