iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IX **Month of publication:** September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74116

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Enhancing DDoS Attack Detection in IoT Networks Through Pruned Deep Ensemble Learning

Asundi Kamal Jason
PG-Scholar, Department of CSE, JNTUA College of Engineering, Ananthapuramu,India

Abstract: ***DDoS attacks prove to be the biggest roadblocks to the maintenance and stability of the IoT environment, occasionally causing network congestions, shutdowns, and breaches of data integrity. In this regard, setting up an intelligent ensemble learning framework for DDoS detection over IoT infrastructures involves pruning of models with an aim of maximizing performance and minimizing computation. The ensemble learning consists of two models first, stacking classifier RF , GB and second, voting classifier of LR, DT, KNN, CNN and LSTM. TPOT Classifier is an automatic approach to model and hyperparameter selection and tuning, maintaining optimized ML pipelines. The features of the dataset include some major network flow feature attributes such as time, the total amount of forward directions of packets, the total amount of backward directions of packets, The packet length forward, the packet length backward, packets backward, statistical length of packet, and the amount of the packet flag. This is very wide scope of factors which is necessary to define the valid traffic. The tendency of pruning (redundancy or less informative) of the ensemble models provides an opportunity of removing the redundant or less informative classifiers, improve computing efficiency without reducing the predictive accuracy. As to the existing measures of performance, The CNN, LSTM outcomes of the experiment that have been described in the paper have shown that the given technique might be effectively employed in an effort to target the identification of DDoS attacks, and the false positives rates would not be high. The high prevalence of DDoS attacks makes the proposed methodology worthwhile in terms of detecting DDoS due to the scalability of the method as well as because IoT networks are limited in resources. The experimental results reflect that fine ensemble training with pruning can show a superior trade-off relative to the traditional machine learning methods; therefore, providing a mature and scaled security methodology in IoT network.***

Keywords: ***DDoS attacks detection in an internet of things, ensemble learning, pruning, Stacking Classifier, Voting Classifier, TPOT Classifier, CNN, LSTM maintenance of security at a network level, machine learning concepts, network flow measurements, efficiency of computation, scale.***

I. INTODUCTION

Recent developments in the IoT have led to the amplification of the connectivity, hence opening the IoT network to diverse cybercrimes, including DDoS attacks. Such attacks severely interfere with services that are offered by the network leading to loss of money, as well as, leakage of sensitive information. The attack and its outcomes cannot be mitigated or detected in real-time when using a normal security mechanism, given the sheer massive data generation aspect of an IoT scenario and resource cons of the IoT devices. It is, therefore, urgent to develop efficient, scalable, and precise detection mechanism to cope with the constraints of the IoT networks. This paper shall work in the direction of coming up with a DDoS detecting technique that is highly accurate and has lower computational cost by combining deep ensemble methods with pruning techniques. The complement between stacking and voting classifiers and auto machine learning suggests a serious need to improve security in the proliferating IoT landscape.

II. LITERATURE REVIEW

M. F. Saiyedand and I. Al-Anbagi et al.[1] proposed deep ensemble learning system with pruning for detecting DDoS attacks in IoT networks. It employs a flow-based traffic analysis module to preprocess network data and then applies pruning to make the model lightweight and efficient for deployment on resource-constrained devices. This approach ensures fast detection, reduced memory use, and reliable performance, providing a practical solution to strengthen IoT security against evolving DDoS threats.

H. Shafique, K. K. Gupta, M. S. Awan, and M. U. Babar et al. [2] In their comprehensive survey, Shafique et al. provide an in-depth examination of various DDoS attack detection mechanisms tailored for IoT networks. The authors categorize the existing detection approaches into signature-based, anomaly-based, and hybrid methods, highlighting the strengths and weaknesses of each. They also discuss the unique challenges posed by IoT environments, such as resource constraints and the diversity of devices .The survey emphasizes the need for adaptive and scalable detection techniques to effectively mitigate DDoS attacks in heterogeneous IoT networks, setting a foundation for future research in this area.

S. Ahmad, A. H. A. Bakar, and M. I. Ali et al. [3] Ahmad and colleagues propose a hybrid model that leverages multiple machine learning techniques for the detection of DDoS attacks in IoT environments. The study integrates various classifiers to enhance the detection minimize and accuracy. By employing feature selection techniques, the authors optimize the model's performance, ensuring it operates efficiently within resource-constrained IoT devices.

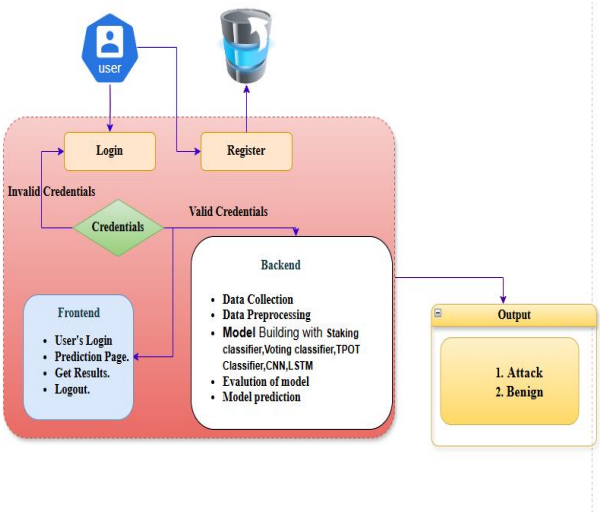
The results demonstrate that their hybrid approach outperforms traditional single-classifier methods, indicating the potential of ensemble techniques in enhancing DDoS attack detection in IoT networks.

M. Al-Mamun, A. I. Abdullah, and M. K. Khan et al. [4] Al-Mamun and co-authors explore the application of ensemble learning methods for DDoS attack detection specifically in IoT networks. Their research emphasizes the importance of combining different learning algorithms to improve detection rates and reduce false alarms. The authors conduct experiments using various ensemble techniques, including bagging and boosting, and assess their effectiveness against DDoS attacks. The findings suggest that ensemble learning not only enhances detection accuracy but also increases the robustness of the detection system against evolving attack vectors, making it a viable solution for securing IoT infrastructures.

W. M. M. Ali, H. S. Abed, and N. A. A. Zawawi et al.[5]

In this study, Ali and his team present an efficient DDoS detection system utilizing deep learning techniques designed for IoT networks. The authors discuss the implementation of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to identify and classify attack patterns from network traffic data. The results indicate that their deep learning-based system achieves high accuracy and low latency, making it suitable for real-time detection in IoT scenarios. The study highlights the potential of deep learning frameworks in enhancing security measures within IoT networks, addressing the complexities of DDoS attacks effectively.

III. ARCHITECTURE DAIGRAM



A. Description

1) Frontend

- Login Page: log in the application and cruisers to register option.
- Registration Page: Allows the newly registered users to link their accounts to insert the required information and establish credibility.
- Home Page: The primary dashboard to the users, through which they can access other functions, including performance Page, the Prediction Page, About Page and the Logout functions.
- Performance Page : Demonstrates the performance values of the predictive models, accuracy and name of the algorithm.
- Prediction Page: Delivers accurate predictions based on the characteristics that can be entered by the user.
- Logout Page: Enables customers to log out of their accounts in a secure manner and close currently open sessions and land on the log in page ready to log in again. Methods Applied

2) Backend

- Data collection: It refers to the systematic process of obtaining information from different sources to support analysis, interpretation, and informed decision-making. It includes determining the type of data required, selecting appropriate methods for gathering it, and maintaining accuracy and reliability throughout the process.
- Data preprocessing : the process of refining raw data by handling missing values, removing inconsistencies, and organizing it into a structured format. This preparation step ensures that the data is ready for effective use in machine learning, analytics, and research tasks.

B. Models

1) Stacking Classifier: A stacking classifier is an ensemble learning method that integrates predictions from several different models to generate a stronger and more reliable final outcome.

2) Voting Classifier: A Voting Classifier is an ensemble machine learning approach that combines the predictions of multiple models and determines the final output based on the majority vote or highest probability among them

3) TPOT Classifier: TPOT is an automated machine learning (AutoML) library in Python that applies genetic programming techniques to optimize and select the most effective machine learning pipelines.

4) CNN: A Convolutional Neural Network (CNN) is a deep learning architecture designed to efficiently detect and learn patterns within data, making it highly effective for tasks involving images, signals, and sequential information

5) LSTM: Long Short-Term Memory (LSTM) is a type of recurrent neural network that regulates the flow of information through input, forget, and output gates, allowing it to retain or discard data across long sequences. This makes it particularly effective for sequential tasks such as speech recognition, natural language processing, and analyzing network traffic.

6) Evalution of model :To determine how accurately a model predicts outcomes on new, unseen data.

7) Model predicition: the process of estimating outcomes or results based on a given model

IV. PROPOSED APPROACH

A. Stacking Classifier

Stacking Classifier is an interfacing algorithm that fuses a number of elementary classifiers in an attempt to create a better prediction. It takes the advantages of the different algorithms that best suit an ultimate estimator the output of which are the base models . The Random Forest and Gradient Boosting models have been used as the base models in this implementation. The algorithms are efficient in identifying complex trend in the data as a result of factors such as they are ensembles and the fact that they accommodate the diversity of feature interaction .The last estimator in such stacking technique is that of the Logistic Regression. It averages the results in basis models and presents a weighted resolution, according to the outcomes of the models, and, hence, enhances overall accuracy.

1) Training Process: The training process follows by training the base models training dataset, that is, learning the patterns and relationships on training data through Random Forest and Gradient Boosting.

2) Feature Generation: The outputs of the base models (their predictions) will be taken as features and used to fit the final estimator (Logistic Regression). This gives the model to learn how to combine the random base models in an optimal way.

3) Prediction: In predictions of new data, The base models make their predictions and then input the estimator. The output of the end result is as a result of grouping of the base models in order to have a decision that is more informed

B. Voting Classifier

Voting Classifier An ensemble method of learning is a form of voting in setting where multiple, different models are given where their output is combined to create a final one. It combines the predictions of most (in classifications) or averages (in regressions) in order to increase overall performance of the forecasting.The 3-base models employed in the current application include Logistic Regression, K-Nearest Neighbor (KNN) and Decision Tree (DT), which is applied by the Voting Classifier. The usability of ensemble of individual algorithms is highly strong due to each providing contributions to be robust and accurate in terms of the outcomes.

1) Training Process: The training process starts by training the three models (Logistic Regression, KNN and DT) using the training dataset

2) Model Prediction: The base models are then trained separately and make individual predictions on new data provided. These values obtained in each model are gathered to be ready to be aggregated.

3) Voting Mechanism: The Voting Classifier combines the predictions of its base learners and makes the final decision using a majority vote approach. In classification problems, prediction is based on a majority decision where by the category with the greatest number of votes across the individual models forms the final prediction
4) Final Results: The prediction cascade of the Voting Classifier indicates a synthesis of the collective models, to yield a more trusted result, than when modest alone. This solution increases the potential of the classifier to work with different types of information and the detection accuracy in the work

C. *TPOT **Classifier***

The TPOT Classifier is an automated machine learning framework that uses genetic programming to design and optimize complete machine learning pipelines. It is effective in the optimization of the search over the optimal models and hyper parameters of given dataset and it also accelerates the process of model selection considerably.

1) Automated Pipeline Creation: TPOT will automatically construct and test a variety of machine learning pipelines (i.e. create a combination of preprocessing techniques, feature selection and model training). This automation assists in establishing the most efficient way of dealing with a given problem.
2) Algorithm Diversity: The TPOT Classifier provides extensive algorithm-specific diversity through a variety of different algorithms used during optimization and a number of distinct tree-based and ensemble models. This variability adds to its capability to identify a proper model in the dataset.
3) Training Process: The training process will start by having TPOT analyze the dataset given and produce possible pipelines. It assesses these pipelines depending on their performance rating in order to come up with the suitable configuration.
4) Feature Transformation: This is also a possibility throughout the pipeline optimization, where TPOT might implement feature transformation, such as scaling and encoding, to improve the model performances. This makes sure that the data is well prepared to train.
5) Final Model Selection: TPOT will attempt to identify the highest-performing model and optimal hyperparameters, based on scores gathered by processing several pipelines. The resulting model can than be deployed, which provides a simplified and efficient solution to the classification problem at hand.

D. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) is a deep learning model that is most often applied in image and video recognition, among others that apply spatial data. It is similar to the visual processing portion of the human brain in that, it extracts features automatically in the input data through the use of convolutional layers.

The CNNs are exceptional and more efficient when it comes to learning spatial hierarchies and this could be used to accomplish image classification tasks.

1) Feature Extraction :The CNNs simplify the task of extracting features manually in the data since the patterns are learned automatically in the hierarchical learning process. They are composed of different layers, including convolutional layers, pooling layers, and fully connected layers.These convolutional layers apply filters (kernels), to the input data, to recognize patterns like edges, textures and shapes, and the pooling layers perform dimension reduction by pooling local properties of the input.
2) Training Process: The CNN is trained via what is known as backpropagation in which the filters of the convolution layers have their weights adjusted relative to error between the estimated and true output. During the training process, the model learns the way through which it can identify several aspects of the raw data presented to it, and in the process, better its chance of making the right predictions.Inference Mechanism
3) In prediction, a new input to the network is fed through its trained CNN model and using the trained filters within the convolution layers, the data is transferred through the fully connected layers to provide an output prediction. This model estimates a class by means of applying a softmax operation to transform the output into probabilities of each type of class.
4) Model Evaluation: The performance of CNNs is evaluated using metrics such as accuracy, precision, recall, and the F1-score. The performance of the model can be enhanced further by tricks such as data augmentation, dropout regularization, and transfer learning. Overfitting can be avoided and generalization to unseen data guaranteed by cross-validation.

E. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an advanced type of recurrent neural network (RNN) designed to effectively capture and manage long-term dependencies within sequential data. LSTMs are at the top of time-series and other types of analysis where there is a requirement to induce long-term dependencies in data, i.e. natural language processing, speech recognition, etc.

1) Processing of discrete Data: An LSTM network is memory cell based and allows the network to retain and maintain information over a long duration of time. Unlike a traditional RNN, LSTMs have the input, forget and output gates that allow controlling whether the information is stored or dismissed in this way it can remember crucial data and forget unnecessary information. The latter capability makes LSTMs suitable in processing sequential data and where the order and the background of such a data is significant.

2) Training Process: The LSTMs training is optimized through backpropagation through time (BPTT) algorithm, which is achieved by updating network weights of the gap between the predicted and actual outputs of the few-time steps. The LSTMs ensure they recollect significant features even on the long sequences and guarantee reduction of the prediction errors whilst optimizing the weights of the LSTMs.

3) Mechanism of Forecast: In prediction, a trained LSTM-based model takes the input sequences one step at a time. Each step transfers input data into the memory cells and modifies the state of the network generating the output data based on the learnable or trained weights. The end product is produced upon processing all time steps and the model then makes a prediction based on the learned patterns in the data.

4) Model Evaluation: The LSTMs are tested based on normal classification measures (i.e. accuracy, precision, recall and F1-score). Optimization of the model could be further done by hyperparameter tuning, dropout, and batch normalization. There is a tendency to apply cross-validation to evaluate model performance and prevent overfitting of complex sequential data.

V. RESULTS AND DISCUSSIONS

A. Imports and App Setup

- Flask imports: For building the web interface and handling HTTP requests.
- MySQL connector: For connecting to your database (users and authentication).
- Pandas & NumPy: For data loading and numerical operations.
- scikit-learn: For preprocessing, traditional ML classifiers, and evaluation metrics.
- TPOT: Automated ML tool to select and optimize classifiers.
- Keras (TensorFlow): To build deep learning models (CNN and LSTM).

B. Flask App & Database Connection

- Creates a Flask app instance.
- It establishes a connection with a local MySQL database called ddos.

C. Load & Preprocess Data

- Loads CSV dataset with network traffic features.
- Cleans column names and standardizes labels.
- Numeric conversion & cleaning

D. Train/Test Split & Scaling

- Splits data into training and test sets.
- Scales features for traditional ML models.
- Keeps separate splits for deep learning models.

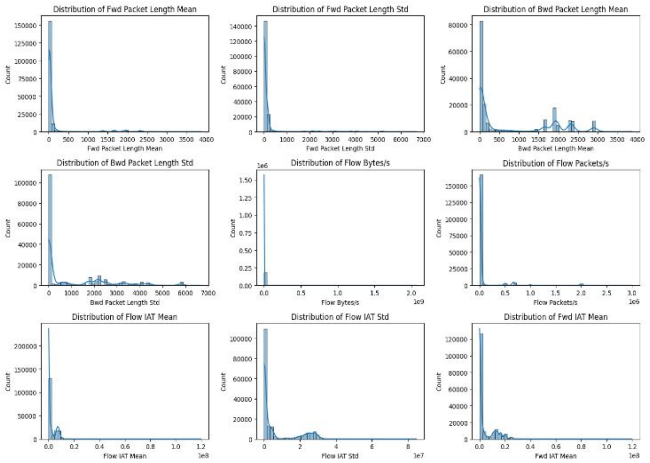
E. Flask Routes

- User Authentication, Prediction, Performance Evaluation

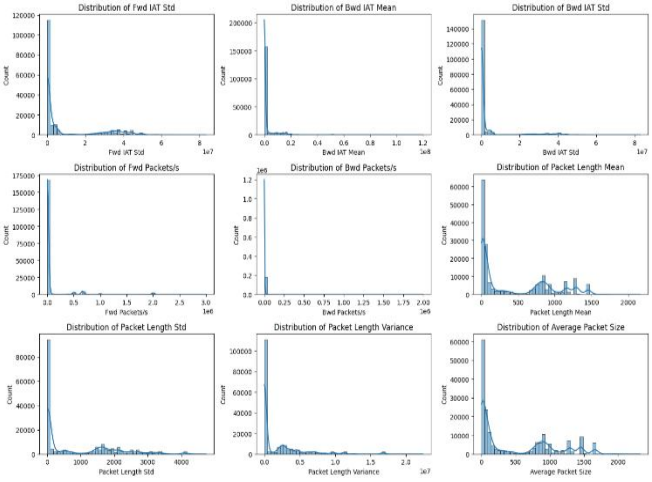
F. Running the app

Exploratory Data Analysis (EDA):

1) Histogram



Most of the distributions can be described as right-skewed, which means that there is a high population of the majority of the data to be at lower values and few high outliers, where most flows could be illustrated with smaller amount of packets length and traffic rates, among which there are a few drastically higher flows. In addition, the majority of the distributions count values are clustered around zero in the case of metrics, i.e., Flow IAT Mean, Flow Bytes/s that data rates related to many flows are sufficiently low. There are also some distributions that have spikes or concentrations at some points such as. Lastly, the Flow IAT Std depicts that the variance of inter-arrival time of the different flows is very large, and having this large variance indicates the presence of both regular and sporadic, perhaps malicious, traffic patterns, and the knowledge on what can cause such variance is valuable in the detection of security anomalies in a network.

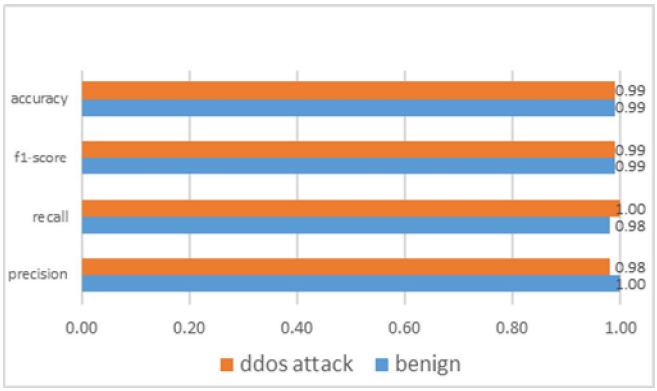


this indicates that most flows could be built on normal conditions whereas there are some flows that could be substantially higher in terms of IAT or packet length. Also most of the counts in all distributions, especially Fwd Packets/s and Bwd Packets/s, are concentrated around the value of zero indicating that most flows have a small number of packets, as would be expected in ordinary network traffic, although larger packet rates are rarer, possibly reflecting activity bursts or some particular application. The distributions of Packet Length Std and Packet Length Variance also show that some of the flows are more uniform in their packet length distribution than others, with some displays being very spiky showing the extreme variation in packets sizes which may point to divergent types of applications or traffic patterns under study, this is critical to bear in mind when performing measurements to ascertain the network performance levels. In addition, the distribution in Fwd IAT Std and Bwd IAT Std are somewhat peaked at the one end, meaning that a lot of packets are sent with minimal delay, whereas the outliers on higher points might indicate some anomaly or anomalous activity in the network, necessitating further examination to achieve network security or optimization.

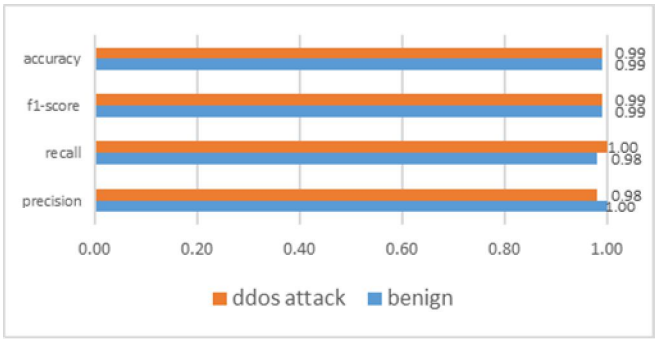
2) Algorithm Comparison

Algorithm	Precision	Recall	F1-Score	Accuracy	Support
Stacking Classifier	1.00 0.98	0.98 1.00	0.99 0.99	99	17995 18005
Voting Classifier	1.00 0.98	0.98 1.00	0.99 0.99	99	17995 18005
TPOT Classifier	0.99 0.97	0.97 1.00	0.98 0.98	98	17995 18005
CNN	1.00 0.91	0.98 1.00	0.95 0.95	96	17995 18005
LSTM	0.74 1.00	1.00 0.64	0.85 0.78	82	17995 18005

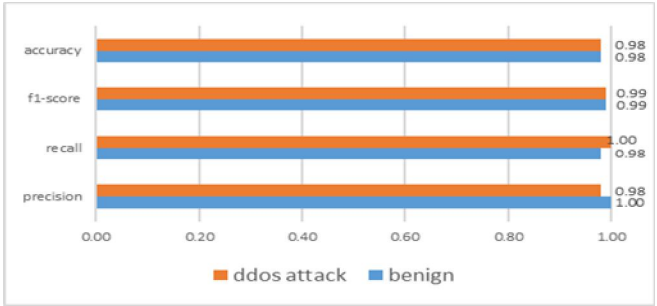
3) Stacking Classifier



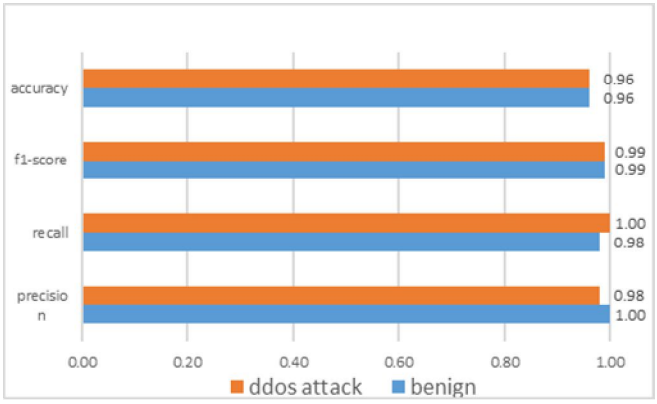
4) Voting Classifier



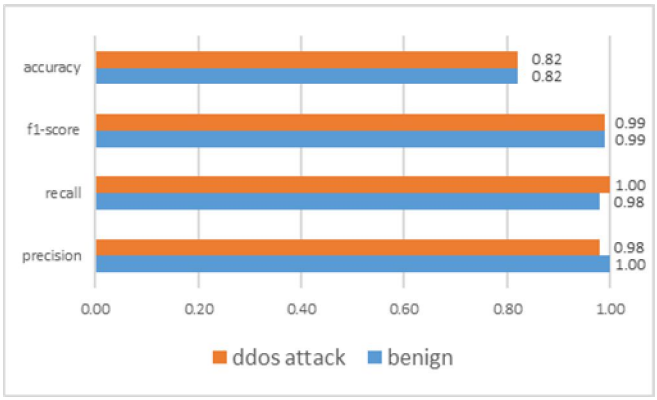
5) TPOT Classifier



6) CNN



7) LSTM



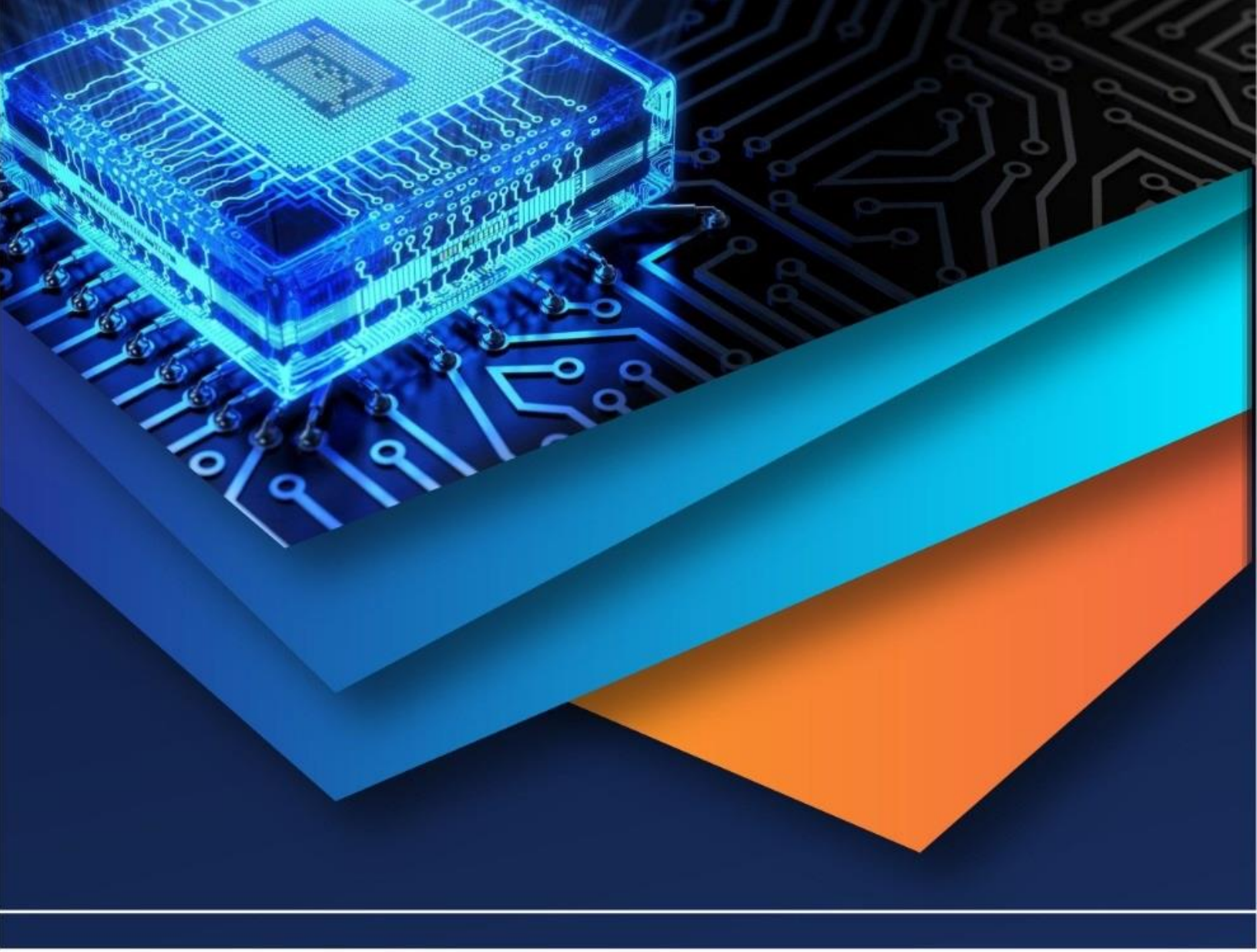
VI. CONCLUSION

The paper suggests the technique that may become quite useful in the identification of the Distributed Denial of Service (DDoS) attack beat in the Internet of Things (IoT) based networks with the concept of the new deep ensemble learning approach. By comparing and combining Stacking Classifier, Voting Classifier and TPOT Classifier in the pipeline Optimization process we achieved a Detection accuracy score of 99 and 98 percent, the latter performing almost equally well. The CNN classifier was somewhat encouraging with a reading of 96 as opposed to that of LSTM which read 82. The calculation performance is complemented with the use of the method of pruning, which deletes models not required, thus improve accuracy in the prediction and efficiency. The effectiveness in separating typical and malicious network flows patterns, reflected in cross-validation and F1 score, may be explained by the high level of feature extraction based on the dataset, and in turn capturing the essential properties of the network flows. It is no surprise that the research findings reveal that deep ensemble model can deliver high detection accuracies bearing in mind that it poses a challenge to the limitations of IoT resources hence contributing to the scaling challenge in the securities of IoT networks amid DDoS attacks. Although this study is also capable of providing an influential contribution to knowledge development on the issue of cybersecurity.

REFERENCES

[1] M. F. Saiyedand and I. Al-Anbagi, “Deep Ensemble Learning With Pruning for DDoS Attack Detection in IoT Networks,” in IEEE Transactions on Machine Learning in Communications and Networking, vol. 2,pp.596-616, 2024,doi:10.1109/TMLCN.2024.3395419.
[2] Shafique, H., Gupta, K. K., Awan, M. S., & Babar, M. U. (2021). A survey of detection mechanisms for Distributed Denial of Service (DDoS) attacks in Internet of Things (IoT) networks. IEEE Access, 9, 102988–103005.https://doi.org/10.1109/ACCESS.2021.3098043
[3] Ahmad, S., Bakar, A. H. A., & Ali, M. I. (2021). A hybrid machine learning-based model for detecting DDoS attacks in IoT environments. Future Generation ComputerSystems,119,41–54. https://doi.org/10.1016/j.future.2021.01.016
[4] Al-Mamun, M., Abdullah, A. I., & Khan, M. K. (2021). Application of ensemble learning for detecting DDoS attacks in IoT networks. Journal of Information Security andApplications,57,102758. https://doi.org/10.1016/j.jisa.2020.102758
[5] Ali, W. M. M., Abed, H. S., & Zawawi, N. A. A. (2022). A deep learning approach to efficient detection of DDoS attacks in IoT networks. IEEE Access, 10, 20645–20659. https://doi.org/10.1109/ACCESS.2022.314294

[6] Verma, R., Srivastava, A. K., & Gupta, R. K. (2022). Deep learning-based approach for DDoS attack detection in IoT networks. Journal of Computer Networks and Communications, 2022, 1–12. https://doi.org/10.1155/2022/2039265
[7] Qureshi, F. A., Alazab, H. H., & Alzahrani, A. J. H. (2021). A review of DDoS attack detection techniques in IoT networks. Computers & Security, 110, 102427. https://doi.org/10.1016/j.cose.2021.102427
[8] Santos, J. L. M. B., Silva, E. M. F. D., & Silva, R. D. S. (2020). An ensemble learning technique for DDoS attack detection in IoT environments. IEEE Latin AmericaTransactions,18(6),1058–1065. https://doi.org/10.1109/TLA.2020.9189155
[9] Ghosh, S. N., Rajan, R. D. S., & Nair, P. D. S. (2021). An extensive review of machine learning algorithms for detecting DDoS attacks in IoT systems. Journal of Network and Computer Applications, 174, 102866. https://doi.org/10.1016/j.jnca.2020.10286
[10] Arshad, M., Hussain, A. W., & Qureshi, A. G. (2021). A comprehensive survey on deep learning methods for DDoS attack detection in IoT networks. Future GenerationComputerSystems,115,23–37. https://doi.org/10.1016/j.future.2020.08.019
[11] Noor, M. I. S. M., Mohd, A. A. H. Y., & Zulkifli, Z. M. Z. (2022). An ensemble learning-based approach for effective detection of DDoS attacks in IoT networks. Journal of King Saud University - Computer and InformationSciences,34(2),187–194. https://doi.org/10.1016/j.jksuci.2019.06.006



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY