J. Electrical Systems 20-11s (2024): 5170-5179

Dr. Preetish Ranjan[1], Dr. Govind Kumar Jha[2], Hare Krishna Mishra[3]

Distributed Denial of Service Attack Detection using Machine Learning and Deep Learning Approach

Journal of Electrical Systems

Abstract: A Distributed Denial of Service (DDoS) attack can disrupt the availability of resources and safety of online services by flooding targeted networks or servers with excessive traffic. This paper presents a dual-model detection system that uses machine learning and deep learning techniques to tackle the complexity of DDoS attacks. The first model comprises of a stacked ensemble of K-Nearest Neighbors (KNN), Decision Tree Classifier, and Logistic Regression. By combining the predictive capabilities of these algorithms, this model increases detection accuracy while identifying a broad range of assault patterns and behaviors. The ensemble model improves the system's resilience and generalizability to DDoS attack type by combining several classifiers. In addition, the second model automatically recognizes complex patterns in network traffic data using deep learning. This methodology is intended to identify subtle, intricate DDoS assault indicators that conventional techniques could miss. It is especially well-suited for real-time detection in a dynamic network, as its deep architecture enables it to process big datasets efficiently.

Keywords: Machine Learning, Deep Learning, DDoS attack, K-Nearest Neighbors (KNN), Machine Learning, EfficientNetB3, Radial Basis Function.

1. Introduction

One of the biggest challenges to stability and security in online service is Distributed Denial of Service (DDoS) attacks. They flood particular networks or servers with enormous loads of traffic in order to make it inaccessible to legitimate users. The spread of online services and a rapid increase in internet traffic have made networks vulnerable to attacks. These attacks are very common in the digital world, and they have a higher frequency and severity in nature [11]. The attack is simply to send a lot of traffic (usually from many sources) towards the target, e.g., a server, a network, or an online service, so that they have a harder time slowing down or even going offline. These attacks are getting commonplace and sophisticated day by day, which makes it difficult for the traditional defense mechanisms that depend on static rules and signatures to cope with the evolving attack techniques. For the purpose of keeping up with an ever-inventive group of threats, machine learning and deep learning have emerged as potent cybersecurity solutions that allow organizations to detect and react swiftly. Looking at a large amount of traffic, based on this line, these methods will use network view to learn benign and malicious behavior patterns and optionally provide more accurate and flexible detection mechanism. To solve this problem, the above-mentioned work comes into existence, which takes care of the growing demand for effective cybersecurity solutions in this digital era, making use of machine learning and deep learning using various features for building a robust system to detect these attacks. Advanced detection systems are needed to properly identify and respond to these attacks, as they continue to become increasingly sophisticated and frequent in real time. This is effective because it takes advantage of traditional machine learning techniques and cutting-edge deep learning models.

The final prediction in this model is made with the help of the meta-classifier, Logistic Regression, when the results of base classifiers are merged. It combines the distinctive strengths of each classifier to give a better accuracy. By taking advantage of the different decision-making ability of these models, we let all these algorithms be our learner and their sensors to capture various patterns and behaviors associated with DDoS attacks, which in turn increases the detection accuracy.

[1]Amity Scool of Engineering and Technology, Amity University, Patna, India
[2] Department of Computer Science & Engineering, Government Engineering College Munger, Bihar, India
[3]Department of Electrical and Electronics Engineering, B.P Mandal College of Engineering Madhepura, Bihar, India
* Corresponding Author: Govind Kumar Jha
E-mail Id: gvnd.jhal@gmail.com

The idea behind this methodology is to efficiently manage large volumes of network data and find anything out of the ordinary that would suggest an ongoing DDoS attack. This work combines implementations of these two approaches in order to represent opportunities for deep learning research and evaluation of machine learning) technologies in the domain of cybersecurity.

2. Literature Review

The stacked model learned many classifiers. The stacked ensemble model also includes K-Nearest Neighbors, Decision Tree Classifier, and Logistic Regression to build a better ensemble model. A total of three classifiers are integrated to build the stacked ensemble model, including K-Nearest Neighbors (KNN), Decision Tree Classifier, and Logistic Regression. A deep learning model was also developed in addition to the previous stacked machine learning model. As deep learning algorithms can automatically extract complex data features, they are ideal for identifying subtle and nonlinear patterns in network traffic.

This methodology had a quality filtering step to remove columns which were highly null and used mean fill for missing values. This is further to improve the model performance by application of PCA (Principle Component Analysis) for important feature extraction and dimensionality reduction[3]. At the correct depth, a deep learning model has been trained to efficiently learn from intricate patterns and yield optimal predictions without overfitting. Further, this approach has been broadened to the Neural Network by adding more layers to the neural network's model using regularization techniques like batch normalization, dropout, and L2 regularization for better generalization [4] . Pattern recognition and some machine learning algorithms are applied, mainly focusing on the importance of LLM 5. The cross-validation and hyperparameter optimizations are carried out on the transformed deep learning models. A meta-classifier (Logistic Regression) is used here to aggregate the outputs of machine learning classifiers and integrate with the outputs of machine learning based models to improve the precision. Karthikeyan et. al. SUMMARY This paper aimed at modeling the WSN-IoT system to improve security and then using machine learning algorithms with the Firefly algorithm over the model to find out its vulnerabilities. Moreover, two dissimilar methods for intrusion detection in WSN-IoT was also combined by them[7]. Jayaraj et. al. focused on phishing URL intrusion detection with Hybrid Ensemble Feature Selection. To cope with the quantification operation made by the score calculation step, they introduce the Cumulative Distribution Function gradient (CDF-g) algorithm to yield a series of subsets of primary features for secondary feature subset generation in the data perturbation ensemble [8]. Anil et. al. Bagging and boosting techniques are used to reduce the false positive rate while feature selection in high dimension dataset have been done using Opposition-based Northern Goshawk Optimization algorithm for intrusion detection [9]. Introducing M-MultiSVM as an efficient way to select sub-classes by means of Support Vector Machine (SVM). Robinson et. al. relevant feature selection using correlation-based and information gain-based methods in training and testing [10]). Most of the recent works only attempted to utilize machine learning and data mining algorithms just to increase the performance (Hatata et al. In this regard, this paper also used K-Nearest Neighbors (KNN), Decision Tree Classifier and Logistic Regression which are more efficient as well.

3. Research Gap

Our objective is to enhance overall predictions using the strengths of each model. Stacking integrates the predictions from multiple base models and utilizes them as input for a final model, termed a meta-classifier. The dual-model approach offers a complete DDoS attack detection solution that is highly accurate and flexible enough to adjust to changing threats. Our goal in doing this study is to help create network security solutions that are more intelligent and resilient.

4. Methodology

Machine and deep learning techniques use a two-pronged approach to detect distributed denial of service threats. The methodology includes the following:
1. Data collection and preprocessing : publicly available datasets, such as network traffic data, are used to train and assess the models on both benign and malicious materials. For certain cases, the models must distinguish between the two.
2. Model creation : two separate models must be created to detect DDoS attacks: a deep learning approach and a stacked ensemble machine learning model that combines the results of many other base classifiers.

3. Performance assessment: the models must be tested on how they perform in real-world scenarios by measuring their accuracy, precision, recall, and F1-score. The stacking model was trained using labeled network traffic information and used each base classifier's trends to generate forecasts. The meta-classifier, Logistic Regression, analyzes the models and determines which one is best for which situation to ensure a more precise and trustworthy forecast. Cross-validation was used to ensure that the model was generalizable and that bias was minimised. To ensure the stacked classifier's reliability and overfitting avoidance, more data was needed to validate. A multilayered deep study model can autonomously study how to acquire complex attributes from network traffic data. The architecture includes dense nodes, dropout nodes, batch normalization nodes, and an output layer for binary classification. In this situation, we added five additional layers to the neural network, making it deeper. This configuration gives it the capacity to improve its ability to detect complicated patterns present in DDoS attacks. The Adam optimizer and backpropagation, which utilizes the binary cross-entropy loss approach, were used to train the deep learning model. Batch Normalization, Dropout, and L2 normalization are additional methods utilized to enhance the model's generalization and mitigate overfitting. The DDoS assault pattern detector explains how these techniques allow the model to capture maximum detection even when confronted with real-world data.

Python is a renowned programming language known for its high proficiency when it comes to data analysis and scientific computing as well as its simple syntax. Due to the vast library ecosystem, it is one of the popular choices for deep learning as well as machine learning models. A powerful and easy to use python machine learning software library for implementing this ensemble technique is Scikit-learn. Pandas is a data analysis and manipulation library. There are also data structures that simplify the direct approach to deal with structured values such as DataFrames and Series. It is used for filtering, transformation and cleaning of the data. NumPy is a Python library used to perform numerical operations whereas Pandas helps in making data processing use cases like filtering, categorization and combining the data.

It also supports performing mathematical functions on these arrays and large, multi-dimensional arrays and matrices. Built and released by Google, TensorFlow is a free and open-source library for the task of designing and training machine learning models, with a heavy emphasis on deep learning. It offers scalability and adaptability to install models on various platforms. Keras is a high-level neural networks API built on top of TensorFlow. It is designed for easy model building and experimentation.

Keras provides a more intuitive way to define complex neural network topologies. A variety of layers, optimization techniques and activation functions will be there. A toggle container for an arbitrarily labelable block of elements in the style of a collapsible notebook, coincidentally named as nbinteract. Matplotlib: Matplotlib is a Python 2D plotting library that produces quality figures and graphs at any IDE. Seaborn: Seaborn is a statistical data visualisation library built on top of Matplotlib. This gives it a superior edge in the interface department when creating visual statistics that are not only visually appealing but highly informative as well.

Three main goals are the focus of this work:

Stacked Machine Learning Structured: It is a combination of an ensemble technique, K-Nearest Neighbors (KNN) + Decision Tree Classifier + Logistic Regression to enhance accuracy and reliability in DDoS attack detection.

Development of Deep Learning Model: A deep learning based detection model that enables the system to detect complex and non-linear attack patterns by automatically extracting complex features out of network traffic data.
Evaluate and Compare Model Performance: Experiments are performed on several real-world datasets to check both models' accuracy, speed, and generalization power.

Researchers at the university hope this will lead to intrusion detection systems (IDS) that are stronger and more adaptive, in turn reducing the impact of DDoS attacks on essential internet services.

5. Results Visualization

The CIC-2019 dataset is used, having labelled network traffic data for benign and malicious instances, including DDoS attacks. This data set has been developed by the Canadian Institute for Cybersecurity at the University of New Brunswick .The sample dataset has been displayed in Fig. 1. The purpose behind selecting this data set is to design a realistic model for modern Distributed
Denial-of-Service (DDoS) scenarios involving reflection and exploitation (e.g., SYN, DNS, LDAP, MSSQL, NetBIOS, SSDP, UDP, UDP-Lag, TFTP).

Destination Port	Flow Duration	Total Fwd Packets	Total Backwards	Total Length of Fwd	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length	Fwd Packet Length Std	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
54865	3	2	0	12	0	6	6	6	0	0	0	0	0	0	0	0	0	BENIGN
55054	109	1	1	6	6	6	6	6	0	0	0	0	0	0	0	0	0	BENIGN
55055	52	1	1	6	6	6	6	6	0	0	0	0	0	0	0	0	0	BENIGN
46236	34	1	1	6	6	6	6	6	0	0	0	0	0	0	0	0	0	BENIGN
54863	3	2	0	12	0	6	6	6	0	0	0	0	0	0	0	0	0	BENIGN
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
80	4877	3	7	26	11601	20	0	8.6	10.26	20	0	0	0	0	0	0	0	DDoS
80	78359838	8	6	56	11601	20	0	7	5.6	20	30018	30018	0	0	0	0	0	DDoS
80	50518	3	6	26	11601	20	0	8.6	10.2	20	0	0	0	0	78300000	78300000	78300000	DDoS
80	78336818	8	6	56	11601	20	0	7	5.6	20	22026	22026	0	0	0	0	0	DDoS

Fig. 1 Tabular dataset containing

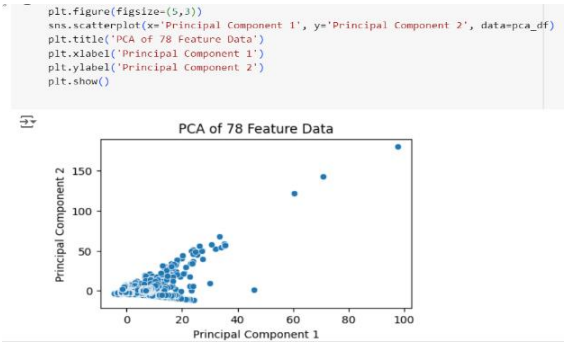
The fig 1 shows a tabular dataset containing network traffic flow features used for classification tasks such as intrusion detection or traffic analysis. Each row in the table represents a single network flow, described by various statistical and packet-level features, and is labeled as either BENIGN or DDoS under the "Label" column. The feature description is as follows:

Destination Port: The network port at the destination side of the communication (e.g., 54865, 55055, 80). This helps identify the application or service in use.
Flow Duration: The total time (in milliseconds or microseconds) for which the network flow was active.
Total Fwd Packets: The total number of packets sent in the forward direction (from source to destination).
Total Backwards: The total number of packets sent in the reverse direction (from destination to source).
Total Length of Fwd: The sum of the sizes of all forward packets.
Total Length of Bwd Packets: The sum of the sizes of all backward packets.
Fwd Packet Length Max / Min / Mean / Std: The maximum, minimum, mean, and standard deviation of packet lengths in the forward direction.
Active Mean / Std / Max / Min: Statistics for active time of the flow, representing the periods when packets were actively transmitted.
Idle Mean / Std / Max / Min: Statistics for idle time, representing the gaps between active transmissions within the flow.
Data Distribution in the Image: The first few rows show BENIGN traffic with relatively small flow durations, fewer packets, and smaller packet length values, indicating normal communication behavior.

The last few rows depict DDoS traffic with certain distinctive patterns, such as very large flow durations (e.g., 78359838), large values in the "Idle Max" column, and higher consistency in packet length values. These patterns are indicative of flooding-type attacks where traffic is repetitive and sustained for a long period.

The difference in statistical features between BENIGN and DDoS entries highlights how such datasets can be used to train machine learning or deep learning models to detect malicious traffic automatically. This data set has 66237 rows and 79 columns, making it high-dimensional. Features are represented by 78 columns, while class is represented by 1. To assure data quality, mean filling and dropping columns with a high percentage of null values have been used to resolve missing values. To choose the most pertinent features for model training and reduce dimensionality, Principal Component Analysis (PCA) has been used. PCA t-SNE can also be used in place of PCA for this purpose. Using Minmax Scaler, the features have been scaled to provide constant input ranges for the models. Using methods like one-hot encoding, string labels have been transformed into numerical values.

Visualisation using PCA :



Several machine learning models were combined using a stacking ensemble technique, which capitalized on each model's unique characteristics to enhance overall predictive performance. This meta-classifier integrated the predictions of the base models (K-Nearest Neighbors, Decision Tree Classifier, and Logistic Regression) to get the final prediction. A linear model for binary classification is called logistic regression. Using one or more predictor variables, it forecasts the likelihood of a binary result. The model produces probabilities by applying a logistic function to a linear set of features. It is the meta-classifier in the stacking ensemble.
By punishing large coefficients, the L2 penalty (ridge regularization) was used to avoid overfitting. In order to regulate the step size during optimization, a learning rate of 0.1 was employed.

A Decision Tree Classifier model bases its judgments on a tree-like structure. It generates nodes and branches to classify the data and divides the dataset into subsets using decision rules based on feature values.A non-parametric classifier called K-Nearest Neighbors classifies a data item according to the majority class of its K nearest neighbors. It is a straightforward and efficient technique that uses the proximity of data points in the feature space to generate predictions. By building and training a multi-layered neural network to automatically learn and extract intricate information from network traffic data, a deep learning technique was used to detect DDoS attacks.

These nodes are organized into separate layers. These two or three-dimensional input data are passed through the layers of neurons to create a neural network. Data can be processed and transformed at each layer as per how they are trained. Network design determines the depth and complexity of this model. Five extra layers were added for the neural network to learn deep down sophisticated patterns and relations in the data.
Backpropagation: The algorithm by which the neural network is trained is a modification of weights and activation layers based on the error gradients calculated after running each image through training. In binary classification, you might use the Binary Cross-Entropy Loss function to measure your performance compared to the label. We use an algorithm optimizer to perform adaptive moment estimation and adjust learning rates to improve model convergence. Batch normalization is employed to normalize the activations and gradients during training, making them perform well. Dropout is a technique used to improve generalization and prevent overfitting from occurring during training by choosing neurons at random and ignoring them. By adding an L2-Regularization, we penalize a model with big weights and hence the model will generalize well to new data as it faces the overfitting issue.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 128)	10,112
batch_normalization_8 (BatchNormalization)	(None, 128)	512
dropout_8 (Dropout)	(None, 128)	0
dense_10 (Dense)	(None, 64)	8,256
batch_normalization_9 (BatchNormalization)	(None, 64)	256
dropout_9 (Dropout)	(None, 64)	0
dense_11 (Dense)	(None, 32)	2,080
batch_normalization_10 (BatchNormalization)	(None, 32)	128
dropout_10 (Dropout)	(None, 32)	0
dense_12 (Dense)	(None, 64)	2,112
batch_normalization_11 (BatchNormalization)	(None, 64)	256
dropout_11 (Dropout)	(None, 64)	0
dense_13 (Dense)	(None, 128)	8,320

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 128)	10,112
batch_normalization_8 (BatchNormalization)	(None, 128)	512
dropout_8 (Dropout)	(None, 128)	0
dense_10 (Dense)	(None, 64)	8,256
batch_normalization_9 (BatchNormalization)	(None, 64)	256
dropout_9 (Dropout)	(None, 64)	0
dense_11 (Dense)	(None, 32)	2,080
batch_normalization_10 (BatchNormalization)	(None, 32)	128
dropout_10 (Dropout)	(None, 32)	0
dense_12 (Dense)	(None, 64)	2,112
batch_normalization_11 (BatchNormalization)	(None, 64)	256
dropout_11 (Dropout)	(None, 64)	0
dense_13 (Dense)	(None, 128)	8,320

```
# Training the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

```
Epoch 1/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 34s 16ms/step - accuracy: 0.7567 - loss: 7.1569 - val_accuracy: 0.9835 - val_loss: 0.8307
Epoch 2/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 51s 22ms/step - accuracy: 0.9656 - loss: 0.6136 - val_accuracy: 0.9848 - val_loss: 0.2875
Epoch 3/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 46s 25ms/step - accuracy: 0.9717 - loss: 0.3146 - val_accuracy: 0.9843 - val_loss: 0.2670
Epoch 4/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 70s 18ms/step - accuracy: 0.9717 - loss: 0.3116 - val_accuracy: 0.9855 - val_loss: 0.2719
Epoch 5/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 38s 16ms/step - accuracy: 0.9759 - loss: 0.2872 - val_accuracy: 0.9841 - val_loss: 0.2523
Epoch 6/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 27s 16ms/step - accuracy: 0.9720 - loss: 0.3055 - val_accuracy: 0.9853 - val_loss: 0.2493
Epoch 7/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 25s 15ms/step - accuracy: 0.9762 - loss: 0.2833 - val_accuracy: 0.9871 - val_loss: 0.2176
Epoch 8/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 44s 17ms/step - accuracy: 0.9750 - loss: 0.2956 - val_accuracy: 0.9846 - val_loss: 0.2145
Epoch 9/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 39s 16ms/step - accuracy: 0.9774 - loss: 0.2713 - val_accuracy: 0.9848 - val_loss: 0.3215
Epoch 10/10
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 25s 15ms/step - accuracy: 0.9792 - loss: 0.2797 - val_accuracy: 0.9983 - val_loss: 0.1886
<keras.src.callbacks.history.History at 0x7d8088fcb9a0>
```

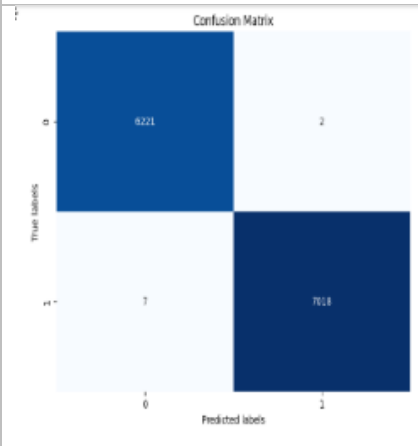
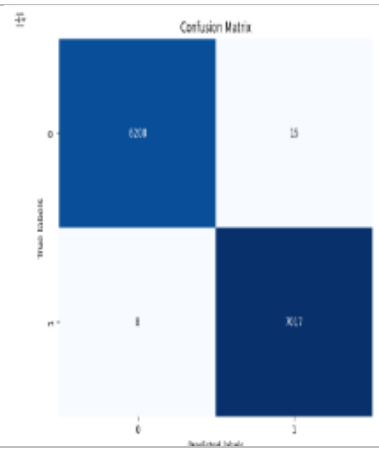
A range of parameters measures the performance of the stacked machine learning and deep learning models. The number of accurate predictions divided by the total number of predictions is the accuracy.

Precision =True Positives / (True Positives + False Positives)
Recall = True Positives / (True Positives + False Negatives)
F1-score = 2 × Precision × Recall / (Precision + Recall)

The stacked ensemble model delivered a high rate of correct predictions being 99.95% accurate with the test data set. This means 99.99% of all positives were truly positive, according to the level of real positive detections throughout all predictions. For the DDoS class, if the recall was 99.93% among all real attacks (including both fresh and well-known attacks), then this demonstrates high model confidence that it can successfully detect DDoS attacks accurately. The F1-score is balanced between precision and recall to produce a performance metric with having value of 99.96% .

In order to identify whether network traffic was malicious or benign, the deep learning model displayed an accuracy rate of 99.83%. The deep learning model showed the accuracy of positive predictions, with a precision value of 99.79%, for DDoS detection. The model's ability to identify actual positive cases is proven by the DDoS detection recall of 99.89%. The 99.84% F1-score shows that the model can detect and classify attacks as needed, balancing between precision and recall.

Comparison

	Stacked Machine Learning Model	Deep Learning Model
Performance	It demonstrated strong performance in classification tasks, leveraging the diverse strengths of Logistic Regression, Decision Tree Classifier, and K-Nearest Neighbors. The meta-classifier effectively combined predictions from the base models, enhancing overall accuracy and robustness.	The deep learning model achieved competitive performance in detecting DDoS attacks with its deep architecture and additional layers. The use of batch normalization, dropout, and L2 regularization helped mitigate overfitting and improve generalization.
Strength	High interpretability of individual base models, effective combination of predictions through stacking.	Ability to learn and extract intricate patterns from network traffic data, suitable for handling high-dimensional data.
Limitation	May be less effective at capturing complex patterns compared to deep learning models.	Requires careful tuning of hyperparameters, potential risk of overfitting if not regularized properly.
Confusion matrix		

The confusion matrix for both models displays True Positives, True Negatives, False Positives, and False negatives.

This is how well each model was able to tell benign traffic apart from DDoS attacks. Both models successfully detected DDoS attacks according to the performance measures; however, the deep learning model has a little higher accuracy and F1-score. The confusion matrix and classification report elaborate on where each model excels or needs work. The performance of the model on different types of network traffic was better visualised with precision, recall, F1-score and support for each class (benign and DDoS).

Given their strong capabilities to detect DDoS attacks, these models identify the potential for use in real-time network security systems. Efficient DDoS attack detection makes it possible to take prompt actions to mitigate the effects of all attacks and protect critical services and infrastructure. The deep learning model, while having an ability to detect complex patterns of attacks, is less interpretable in how it makes predictions than the stacking ensemble model, which provides a reliable and interpretable solution. In future works, more advanced improvements could be added, including considering other features, testing with other ML and DL architectures, or applying the models to different network attacks. You need to adapt and improve continuously to stay ahead of evolving attack tactics. The findings illustrate the effectiveness of adopting a versatile fusion that leverages advanced deep learning techniques from cybersecurity applications together with traditional machine learning systems. This approach makes the detection process more accurate, and by providing feasible solutions to scale, it reduces the complexity in both detection and elevates the network security domain.

6. Conclusion

This study identified distributed denial of service (DDoS) attacks using deep learning and stacked machine learning models. Stacked Machine Learning Model: KNN, Decision Tree Classifier and Logistic Regression (Ensemble technique). Logistic Regression was used as meta-classifier to aggregate the base models' outputs. The model performed good with accuracy, precision, recall and F1-score. The classification report and confusion matrix show how it distinguishes between a real attack and legitimate network traffic. A deep neural network with multiple layers, such as dropout layers, batch normalization, a dense layer, and an output layer for binary classification was created.Here, two methods were incorporated to recognize Distributed Denial of Service (DDoS) attacks: a deep learning model and a layered machine learning model. Stacked Machine Learning: Stacked Machine Learning Model, as discussed is an ensemble learning methodology and it comprised of K-Nearest Neighbors (KNN), Decision Tree Classifier and Logistic Regression. Logistic Regression was employed to merge the single-class probabilities from the basic models as the meta-classifier. And, the model gave pretty good accuracy, precision, recall and F1-score.

It showed itself to be a good performer discriminating between malicious and benign samples in the classification report and confusion matrix. A deep neural network composed of multiple dropout layers, batch normalization, a fully connected layer, and an output layer was built.

It achieved competitive accuracy and F1-score in comparison to the stacked machine learning model. Use of regularisation techniques such as batch norm, dropout and L2 helped reduce overfitting and improve generalization. We compared this model with another LR model for other performance criteria, such as accuracy, precision, recall, F1-score by doing cross-validation on both models. Results revealed that they can, in fact, detect DDoS attacks, and the deep learning model yields better results in some aspects as well. We hope this sparks enhanced engineering approaches and domain-related feature generation in future research efforts.

Domain-specific knowledge could be incorporated in an effort to improve model performance. Bounding the hyperparameters for deep learning models and the stacking ensemble could enhance performance.This would be implemented by any of the random search or grid search methods. Homogenization: Since information is extracted and utilized, the models must be trained again and undergo constant evaluation to maintain their relevance and performance. The model could be extended to detect network threats other than DDoS, like malicious flow or breaches. If you improve the overall security profile of your network, it can only help. Diverse datasets would enhance the generalizability and robustness of the models. An in-depth characterization of network threats can be realized by amalgamating reports from multiple views and threat categories. The results prove its applicability and give ground for further improvements as well as usage in network security.

Author Contributions

Preetish Ranjan: Conceptualization, Implementation, Software, Field study

Govind Kumar Jha : Methodology, Writing-Original draft preparation, Software, Visualization, Investigation, Validation.

Hare Krishna Mishra: Field study, Data curation, Writing, Reviewing, and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

1. Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials, 15(4), 2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127

2. Rezaei Barzani, A., Pahlavani, P., & Ghorbanzadeh, O. (2023). Ensembling of decision trees, KNN, and logistic regression with soft-voting method for wildfire susceptibility mapping. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 10, 647-652.
3. Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. IEEE Communications Surveys & Tutorials, 21(3), 2224–2287. https://doi.org/10.1109/COMST.2019.2904897
4. Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
6. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer
7. M. Karthikeyan, D.Manimegalai, Karthikeyan Rajagopal, "Firefy algorithm-based WSN-IoT security enhancement with machine learning for intrusion detection", scientific report published in Nature in Jan 2024.
8. R. Jayaraj, A. Pushpalatha, K. Sangeetha, T. Kamaleshwar, S. Udhaya Shree, Deepa Damodaran, "Intrusion detection based on phishing detection with machine learning" Measurement: Sensors, Volume 31, February 2024.
9. Anil V Turukmane, Ramkumar Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning", Computers & Security, Volume 137, February 2024.
10. R. R. Rejimol Robinson, K. P. Anagha Madhav, Ciza Thomas, "Improved minority attack detection in Intrusion Detection System using efficient feature selection algorithms", Expert Systems, Feb 2024.
11. Sulaiman Muhammed Sulaiman, Adnan Mohsin Abdulazeez, "Leveraging of Gradient Boosting Algorithm in Misuse Intrusion Detection using KDD Cup 99 Dataset", Indonesian Journal of Computer Science (IJCS) Vol. 13 No. 5, 2024.
12. Hanaa Attou, Azidine Guezzaz, Said Benkirane, Mourade Azrour, Yousef Farhaoui, "Cloud-Based Intrusion Detection Approach Using Machine Learning Techniques", Big Data Mining and Analytics, 2023.
13. K. Azarudeen, Dasthageer Ghulam, G. Rakesh, Balaji Sathaiah and Raj Vishal, "Intrusion Detection System Using Machine Learning by RNN Method", E3S Web Conf, 2024.
14. Easa Alalwany and Imad Mahgoub, "An Effective Ensemble Learning-Based Real-Time Intrusion Detection Scheme for an In-Vehicle Network", MDPI, 2024.
15. Iacovos Iannou, Palani Murugan, "GEMLIDS-MIOT: A Green Effective Machine Learning Intrusion Detection System based on Federated Learning for Medical IoT network security hardening", ScienceDirect, 2024.
16. G. Logeswari, S. Bose, T. Anitha, "An Intrusion Detection System for SDN Using Machine Learning", Tech Science Press, 2022.
17. N Ojha, A Kumar, N Tyagi, P Ranjan, A Vaish, "Use of machine learning in forensics and computer security", Artificial Intelligence and Cyber Security in Industry 4.0, 211-236.
18. P Ranjan, V Singh, P Kumar, S Prakash, "Models for the detection of malicious intent people in society", International Journal of Digital Crime and Forensics (IJDCF) 10 (3), 15-26.
19. Datasets:
    CICIDS 2017 Dataset. (2019). Canadian Institute for Cybersecurity Retrieved from Github
20. Tools and Libraries:
    Scikit-learn developers. Scikit-learn: Machine Learning in Python. Retrieved from https://scikit-learn.org/
    TensorFlow developers. TensorFlow: An end-to-end open source platform for machine learning. Retrieved from https://www.tensorflow.org/
    Pandas developers. Pandas: Data analysis and manipulation tool. Retrieved from https://pandas.pydata.org/
    NumPy developers. NumPy: The fundamental package for scientific computing with Python. Retrieved from https://numpy.org/
    Matplotlib developers. Matplotlib: Visualization with Python. Retrieved from https://matplotlib.org/
    Seaborn developers. Seaborn: Statistical data visualization. Retrieved from https://seaborn.pydata.org/

Authors' Profiles

Dr. Preetish Ranjan is an Assistant Professor in the Computer Science and Engg. Department at Amity University Patna. He received his Ph.D from IIIT Allahabad. His research area is the implementation of data mining in social network analysis, call data record analysis, recommender systems, and VAPT in network infrastructure. He has published several papers in Scopus and SCI-indexed journals.

Dr. Govind Kumar Jha works as an Assistant Professor and Head of the Department of Computer Science and Engineering at Government Engineering College Munger (Bihar) . He received his M.Tech. and Ph.D from Dr. APJ Abdul Kalam Technical University, Lucknow, India. He has over 15 years of teaching and administrative experience with reputed universities/institutes. He has published various research papers in national & international journals and conferences. His research areas are Recommender Systems and Machine Learning. He is a Lifetime Member of the Computer Society of India.

Hare Krishna Mishra is pursuing his Ph.D. from SLIET, Sangroor. He completed his M. Tech. from NIMS University, Jaipur. He is an Assistant Professor in the Department of Electrical and Electronics Engineering, B.P Mandal College of Engineering, Madhepura. He has publications in various journals and conferences of national and international repute.