PROJECT ON

A MACHINE LEARNING BASED APPROACH FOR CLASSIFYING AND PREDICTING DDoS ATTACKS ALONG WITH BOTNET PREVENTION

ABSTRACT

A Distributed Denial-of-Service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. A collection of such exploited machines (bots), called a botnet, can include computers and other networked resources such as Internet of Things (IoT) devices. DDoS attacks block the access of genuine users to a service for a certain period. Attackers induce malware to the healthy devices by sending malicious Uniform Resource Locators (URLs) to the users. A huge amount of traffic is generated by the bot Personal Computers (PCs) being launched to the target system. With the help of these botnets, the attackers send requests to the target servers. With the increase in the population of bots, the severity of DDoS attacks also increases. This work employs Machine Learning (Naive Bayes, Random Forest, XGBoost) and Deep Learning (DNN) to classify and predict DDoS attacks. Additionally, a Botnet Prevention feature, implemented using Logistic Regression, detects phishing URLs to safeguard devices from being part of botnets. These approaches aim to reduce the damage caused by DDoS attacks to various systems.

INTRODUCTION

A distributed denial-of-service (DDoS) attack occurs when multiple systems flood the resources of a targeted system, such as web servers. These attacks exploit networks of infected computers and IoT devices, forming botnets controlled remotely by an attacker. Each bot in the botnet sends requests to overwhelm the target's IP address, causing a denial-of-service to normal traffic. Different types of DDoS attacks include SYN floods, UDP floods, HTTP floods, Ping of death, ICMP floods, Smurf attacks, Fraggle attacks, and NTP amplification attacks.

DDoS attacks have evolved into sophisticated activities, impacting organizations significantly. Notable attacks on GitHub and Google have demonstrated the destructive potential of such attacks. DDoS attacks can result in revenue loss, erosion of consumer trust, financial compensation, and long-term reputation damage.

To counter DDoS attacks, advanced defense strategies are necessary. Machine learning (ML) and deep learning (DL) techniques are used to analyze network traffic and detect malicious patterns associated with DDoS attacks. Collaborative efforts among security professionals, internet service providers, and organizations are crucial to develop effective countermeasures.

DDoS attacks exploit the normal workings of network services, making them difficult to combat. They can cause prolonged impacts on websites and businesses, leading to financial and reputational losses. Implementing robust defense mechanisms, investing in network infrastructure, and using ML and DL approaches can reduce the impact of DDoS attacks. Ongoing research and collaboration are essential to stay ahead of evolving attack techniques and enhance preventative measures.

To help reduce the damage caused by DDoS attacks to various systems, Machine Learning and Deep Learning approaches for classifying and predicting DDoS attacks are being used in this work. For this purpose, implementation of Naive Bayes, Random Forest and XGBoost classification algorithms as a part of Machine Learning and Deep Neural Networks (DNN) as a part of Deep Learning has been done. Additionally, a feature called Botnet Prevention has been included which detects Phishing URLs to prevent a healthy device from being a part of the botnet. It is implemented by using the Machine Learning technique of Logistic Regression.

RELATED WORK

The paper [1] introduces the PSO-XgBoost model to optimize Network Intrusion Detection Systems (NIDS) accuracy. By combining PSO and XgBoost, the model performs well in multi-classification tasks. Evaluation using the NSL-KDD dataset demonstrates its superiority in precision, recall, macro-average, and mean average precision compared to other models.The paper [2] addresses low accuracy and feature engineering challenges in intrusion detection. The BAT-MC model combines BLSTM, attention mechanism, and multiple convolutional layers for network traffic classification. It effectively captures key and local features using attention and convolutional layers, respectively. The softmax classifier improves performance compared to traditional methods.The paper [3] proposes an ensemble learning model for intrusion detection. It combines decision tree, random forest, kNN, and DNN classifiers using an adaptive voting algorithm. The MultiTree algorithm achieves 84.2% accuracy by adjusting training data proportions and employing multiple decision trees. The adaptive voting algorithm further improves accuracy to 85.2%.The paper [4] evaluates supervised machine learning classifiers in network security algorithms using established and new intrusion detection datasets. It emphasizes the underutilization of contemporary databases and provides insights into performance evaluation for network security tasks.The paper [5] integrates deep learning into NIDS by proposing a bidirectional GRU-based model with hierarchical attention mechanisms. It treats intrusion activity as a time-series event, improving detection by identifying significant features with feature-based and slice-based attention mechanisms.The paper [6] presents a hybrid intrusion detection system that combines the CFS-DE feature selection algorithm with a weighted stacking classification algorithm. CFS-DE reduces dimensionality by selecting an optimal feature subset, while the weighted stacking algorithm improves classification performance by assigning higher weights to well-performing base classifiers.The paper [7] introduces a Machine Learning-based Phishing URL detection system using the Random Forest algorithm. It achieves high accuracy (97.98%), language independence, real-time execution, and feature-rich classifiers. However, custom dataset construction and additional training time are required.The paper [8] proposes a botnet detection method using flow summary and graph sampling with machine learning algorithms. It effectively detects botnet traffic, including unknown botnets, by considering timing patterns and utilizing graph sampling technology. Feature selection and parameter optimization are suggested for further improvements.

MODULE DESCRIPTION

BLOCK DIAGRAM

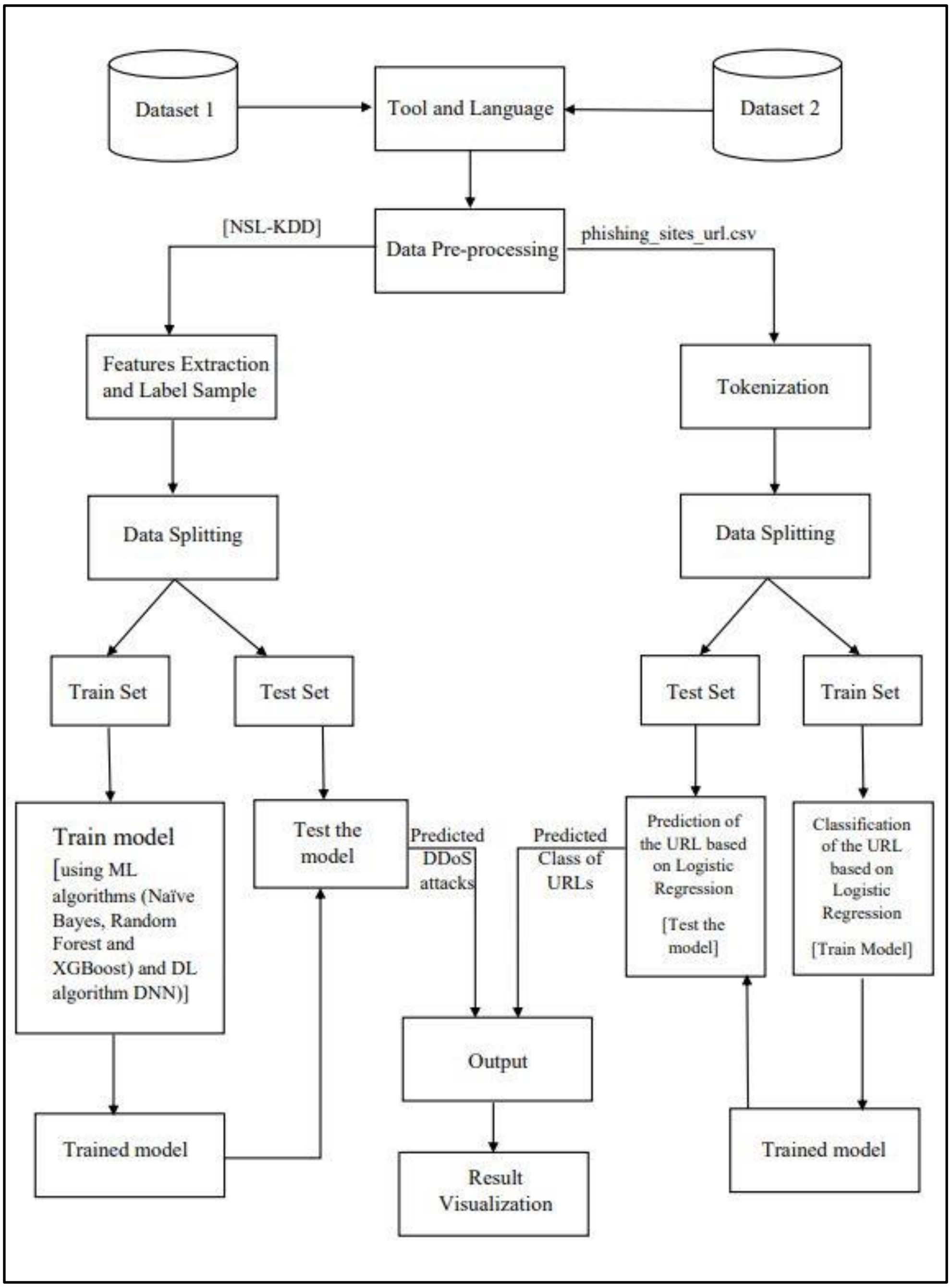

Figure 1:Block Diagram

CLASSIFICATION AND PREDICTION OF DDOS ATTACKS

DATA PREPROCESSING

The data preprocessing involves feature extraction which refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data. The basic aims of feature engineering are to provide an input dataset that is compatible with the criteria of machine learning and artificial intelligence models. As a result, we begin by converting all classified attributes into equivalent numerical labels. The second goal and objective is to improve the performance of machine learning and artificial intelligence models.

INPUT: Dataset

OUTPUT: Processed Dataset

ALGORITHM:

Step 1: Start

Step 2: Take input as Dataset

Step 3: Implement Label Encoding

Step 4: Perform Data Visualization

Step 5: Carry out Feature Scaling

Step 6: Get output as Processed Dataset

TRAINING THE MODELS

Initially, we train the model using some percentage of the dataset for each algorithm. Based on the accuracy, we vary the training percentage of the model

- **NAÏVE BAYES**

The Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

$$P(A/B) = P(B/A)P(A)/P(B)$$

- **RANDOM FOREST**

A random forest algorithm is a collection of decision trees. This algorithm is one of the most popular and powerful machine learning classification algorithms and is used for reaching a lot of decisions in the proposed model.

- **XGBOOST**

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

- **DEEP NEURAL NETWORKS (DNN)**

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships. The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks. We have an input, an output, and a flow of sequential data in a deep network.

INPUT: Processed Train Dataset

OUTPUT: Trained Model

ALGORITHM:

Step 1: Start

Step 2: Take input as Processed Training Dataset

Step 3: Implement Machine Learning and Deep Learning Algorithms

Step 4: Get output as Trained Model

TESTING THE MODELS

After training, we test the model using the remaining percentage of the dataset for each algorithm. Based on the accuracy, we vary the testing percentage of the model. Various performance metrics such as accuracy, precision, F1 score and recall are calculated and plotted.

INPUT: Processed Test Dataset

OUTPUT: DDoS Attack Predicted

ALGORITHM:

Step 1: Start

Step 2: Take input as Processed Testing Dataset

Step 3: Execute Machine Learning and Deep Learning Algorithms

Step 4: Get output as result of DDoS Attack Prediction

Step 5: Stop

BOTNET PREVENTION

DATA PREPROCESSING

The data preprocessing involves tokenization which is the process where the text gets split into words and an array of tokens/words are formed. The system contains string input, so a text tokenizer is used to split the entire input into smaller units called tokens and form an array of tokens. Text vectorizer is used to convert text data into numerical vectors. CountVectorizer converts a given text into a vector based on the frequency (count) of each word that appears throughout the text.

INPUT: URL String

OUTPUT: Array of Tokens

ALGORITHM:

Step 1: Start

Step 2: Take input as URL String

Step 3: Tokenize the URL using RegEx '[A-Za-z]+'

Step 4: Get output as Array of Tokens

CLASSIFICATION OF URL BASED ON LOGICAL REGRESSION (TRAINING)

The classification algorithm in the logistic regression model is to predict the chance of a classification rule. The dependent variable in logistic regression is a binary classification problem that contains data that is coded as 1 or 0. We train the model with some percentage of the dataset based on the accuracy achieved.

INPUT: URL input

OUTPUT: Class of URL

ALGORITHM:

Step 1: Start

Step 2: Take input as URL

Step 3: Perform Classification using Logistic Regression

Step 4: Get output as Class of URL

PREDICTION OF URL BASED ON LOGISTIC REGRESSION (TESTING)

Once the model is trained, we test the model using the remaining percentage of the dataset for each algorithm. Based on the accuracy, we vary the testing percentage of the model. Various performance metrics such as accuracy, precision, F1 score and recall are calculated and plotted.

INPUT: URL input

OUTPUT: Legitimacy of URL Predicted

ALGORITHM:

Step 1: Start

Step 2: Take input as URL

Step 3: Perform Prediction using Logistic Regression

Step 4: Get output whether the URL is legitimate or not to perform botnet prevention

Step 5: Stop

RESULTS AND IMPLEMENTATION

CLASSIFICATION AND PREDICTION OF DDOS ATTACKS

DATA PREPROCESSING

For the dataset that is taken, data preprocessing steps like checking for null values, duplicated rows, were done. The categorical features were identified and converted to numerical data using one-hot encoding technique. Feature Scaling and Feature Selection were carried out to select the best features.

- Standard deviation is checked for 1

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1.
1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1.
1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 1.
1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1.
1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1.
1.
 1. 1.]
```

- Feature Selection:
    - Univariate Feature Selection using ANOVA F-test

```
Features selected for DoS: ['logged_in', 'count', 'rerror_rate',
'same_srv_rate', 'srv_serror_rate', 'dst_host_count', 'dst_host_srv_count',
'dst_host_same_srv_rate', 'dst_host_serror_rate',
'dst_host_srv_serror_rate', 'service_http', 'src_bytes_S0', 'src_bytes_SF']
Features selected for Probe: ['logged_in', 'diff_srv_rate', 'srv_count',
'dst_host_srv_count', 'dst_host_diff_srv_rate',
'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate',
'dst_host_rerror_rate', 'dst_host_srv_rerror_rate', 'Protocol_type_icmp',
'service_eco_i', 'service_private', 'src_bytes_SF']
Features selected for R2L: ['dst_bytes', 'flag', 'hot',
'num_failed_logins', 'is_guest_login', 'dst_host_srv_count',
```

```
'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate',
'service_ftp', 'service_ftp_data', 'service_http', 'service_imap4',
'src_bytes_RSTO']
Features selected for U2R: ['urgent', 'hot', 'root_shell',
'num_file_creations', 'num_shells', 'srv_diff_host_rate', 'dst_host_count',
'dst_host_srv_count', 'dst_host_same_src_port_rate',
'dst_host_srv_diff_host_rate', 'service_ftp_data', 'service_http',
'service_telnet']
```

The attack types are classified under each of the four classes as follows:

- **DDoS (Distributed Denial of Service):** The attack types neptune, back, land, pod, smurf, teardrop, mailbomb, apache2, processtable, udpstorm, and worm fall under the category of "DDoS." These attacks aim to disrupt or deny access to a targeted system or network by overwhelming it with a high volume of traffic or resource-intensive activities. The reason for their classification as DDoS attacks is their nature of flooding the target with excessive requests or data, causing service disruptions or system unavailability.
- **Probe:** The attack types ipsweep, nmap, portsweep, satan, mscan, and saint are classified as "Probe." These attacks involve scanning or probing a target system or network to gather information about its vulnerabilities, open ports, or available services. The attackers use these techniques to assess potential entry points or weaknesses in the target's security defenses.
- **R2L (Remote to Local):** The attack types ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, and httptunnel are categorized as "R2L." These attacks involve unauthorized attempts to gain access to a target system from a remote location. The attackers try to exploit vulnerabilities in services or protocols to bypass security controls and gain unauthorized access to sensitive information or resources.
- **U2R (User to Root):** The attack types buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, and xterm are classified as "U2R." These attacks involve attempts to escalate privileges and gain administrative or root-level access on a target system. The attackers exploit vulnerabilities in software, execute malicious code, or manipulate system resources to elevate their privileges and gain control over the compromised system.

TRAINING THE MODELS

Initially, we train the model using some percentage of the dataset for each algorithm. Based on the accuracy, we vary the training percentage of the model.

NAÏVE BAYES

The Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

$$P(A/B) = P(B/A)P(A)/P(B)$$

GaussianNB
GaussianNB()

RANDOM FOREST

A random forest algorithm is a collection of decision trees. This algorithm is one of the most popular and powerful machine learning classification algorithms and is used for reaching a lot of decisions in the proposed model.

RandomForestClassifier
RandomForestClassifier(random_state=0)

XGBOOST

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

```
XGBClassifier
XGBClassifier(base_score=None, booster=None,
callbacks=None,
              colsample_bylevel=None,
colsample_bynode=None,
              colsample_bytree=None,
early_stopping_rounds=None,
              enable_categorical=False,
eval_metric='error', feature_types=None,
              gamma=None, gpu_id=None,
grow_policy=None, importance_type=None,
              interaction_constraints=None,
learning_rate=0.1, max_bin=None,
              max_cat_threshold=None,
max_cat_to_onehot=None,
              max_delta_step=None,
max_depth=2, max_leaves=None,
              min_child_weight=None,
missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None,
num_parallel_tree=None,
              predictor=None,
random_state=None, ...)
```

DEEP NEURAL NETWORKS (DNN)

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships. The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification. We restrict ourselves to feed forward neural networks. We have an input, an output, and a flow of sequential data in a deep network.

```
2107/2107 [==============================] - 685s 325ms/step - loss: 0.0235 - accuracy: 0.9986

KerasClassifier
KerasClassifier(
      model=<function createModel at 0x7fa857abb9a0>
      build_fn=None
      warm_start=False
      random_state=None
      optimizer=rmsprop
      loss=None
      metrics=None
      batch_size=None
      validation_batch_size=None
      verbose=1
      callbacks=None
      validation_split=0.0
      shuffle=True
      run_eagerly=False
      epochs=1
      class_weight=None
)
```

TESTING THE MODELS

After training, we test the model using the remaining percentage of the dataset for each algorithm. Based on the accuracy, we vary the testing percentage of the model.

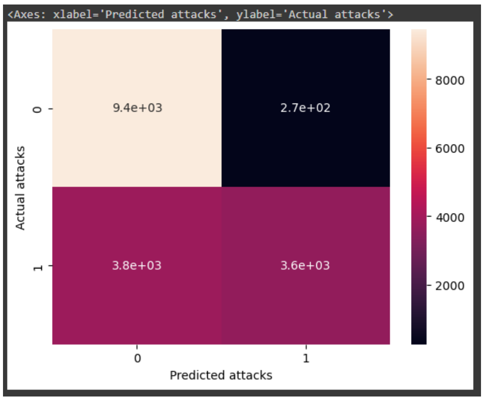

Figure 2: Confusion Matrix-Dos-Naïve Bayes

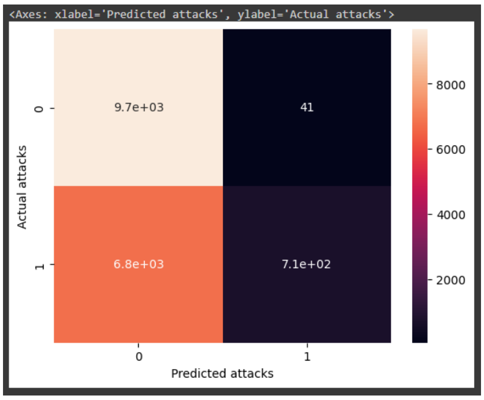

Figure 3: Confusion Matrix-Dos-Random Forest

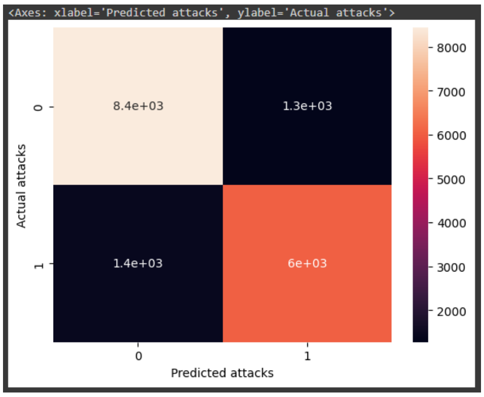

Figure 4: Confusion Matrix-Dos-XGBoost

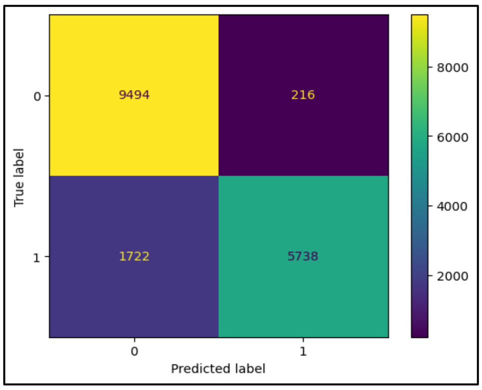

Figure 5: Confusion Matrix-Dos-DNN

RESULT VISUALIZATION

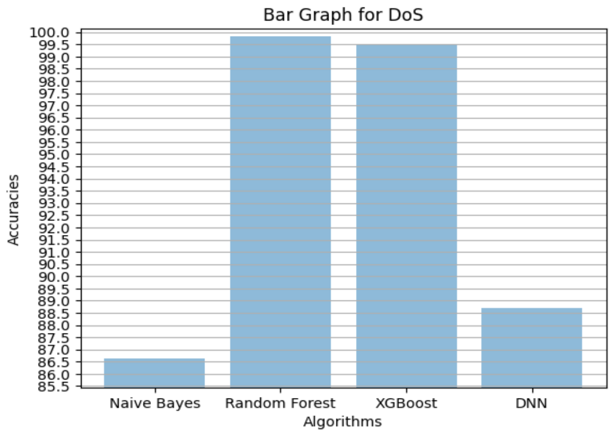

Figure 6:Accuracies vs Algorithms-Dos

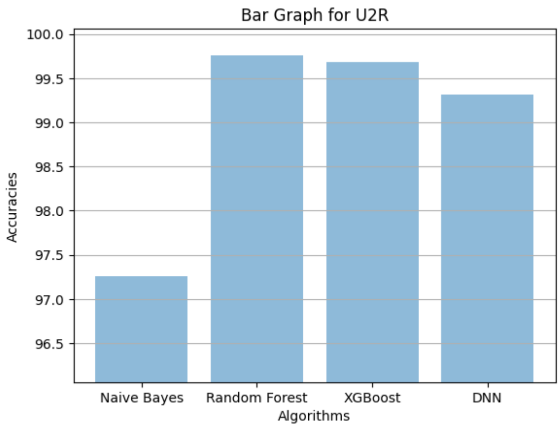

Figure 7:Accuracies vs Algorithms-U2R

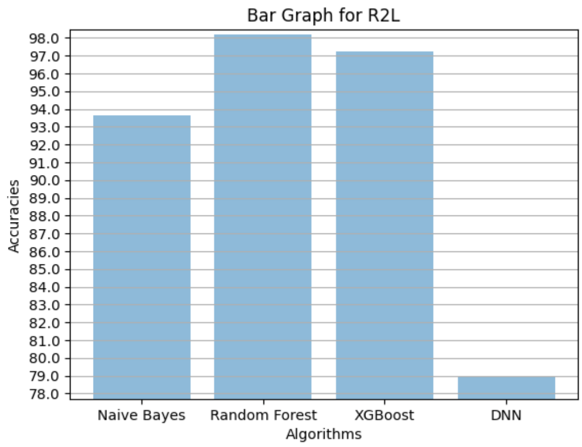

Figure 8:Accuracies vs Algorithms-R2L

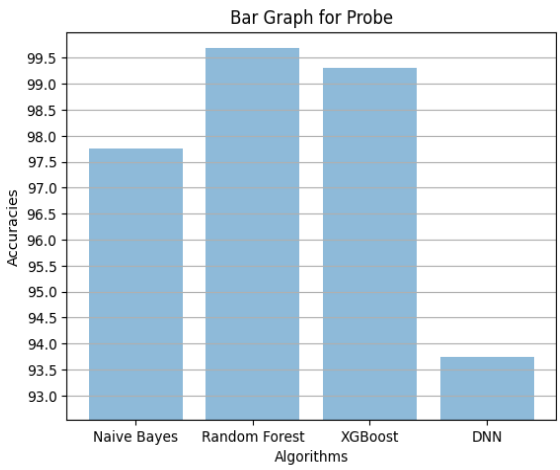

Figure 9:Accuracies vs Algorithms-Probe

PERFORMANCE METRICS:

Table 1:Performance Metrics

Attacks / Algorithms	**Naïve Bayes**	**Random Forest**	**XGBoost**	**DNN**
DoS	86.65%	99.84%	99.51%	88.71%
Probe	97.76%	99.69%	99.301%	93.7%
U2R	97.26%	99.77%	99.68%	99.31%
R2L	93.61%	98.16%	97.21%	78.9%

- For DoS class,
    - The highest accuracy is given by Random Forest Algorithm
- For Probe class,
    - The highest accuracy is given by Random Forest Algorithm
- For R2L class,
    - The highest accuracy is given by Random Forest Algorithm
- For U2R class,
    - The highest accuracy is given by Random Forest Algorithm

BOTNET PREVENTION

DATA PREPROCESSING

The data preprocessing steps involve checking for null values,duplicate values etc.

- Removal of duplicates

	URL	Label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	bad
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	bad
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	bad
3	mail.printakid.com/www.online.americanexpress....	bad
4	thewhiskeydregs.com/wp-content/themes/widescre...	bad
...	...	...
507190	23.227.196.215/	bad
507191	apple-checker.org/	bad
507192	apple-iclods.org/	bad
507193	apple-uptoday.org/	bad
507194	apple-search.info	bad

507195 rows × 2 columns

- Visualizing Target columns

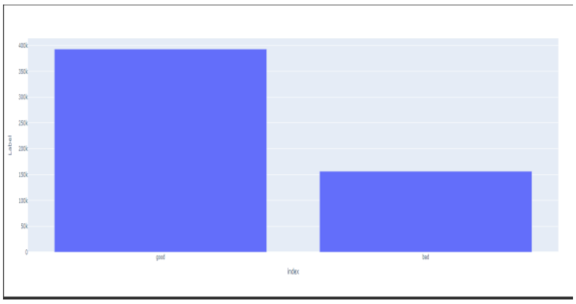

Figure 10: Graph in comparison of count of good URLs and bad URLs

- Handling Stopwords
  Stop words are a set of commonly used words. Proper handling of stopwords in URLs is carried out.

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours',
'ourselves', 'you', "you're", "you've", "you'll",
"you'd", 'your', 'yours', 'yourself', 'yourselves',
'he', 'him', 'his', 'himself', 'she', "she's", 'her',
'hers', 'herself', 'it', "it's", 'its', 'itself',
'they', 'them', 'their', 'theirs', 'themselves',
'what', 'which', 'who', 'whom', 'this', 'that',
"that'll", 'these', 'those', 'am', 'is', 'are', 'was'
```

- Tokenization
  - Tokenization is the process where the text gets split into words and an array of tokens/words are formed. The system contains string input, so a text tokenizer is used to split the entire input into smaller units called tokens and form an array of tokens. Text vectorizer is used to convert text data into numerical vectors.

URL	Label	clean_url	text_tokenized
Carolinarailhawks .com/index.php? id=111	good	Carolinarailhawks .com/index.php?id=111	[carolinarailhawks, com, index, php, id]
Huffingtonpost .com/alex-remington/ ranking-base...	good	huffingtonpost.com/ alex-remington/ranking...	[huffingtonpost, com, alex, remington, ranking...

ckuik.com/ Bobby_Jarzombek	good	ckuik.com/ Bobby_Jarzombek	[ckuik, com, Bobby, Jarzombek]
diafox.xyz/Panel/	bad	diafox.xyz/Panel/	[diafox, xyz, Panel]
chezbob.com/	good	chezbob.com/	[chezbob, com]

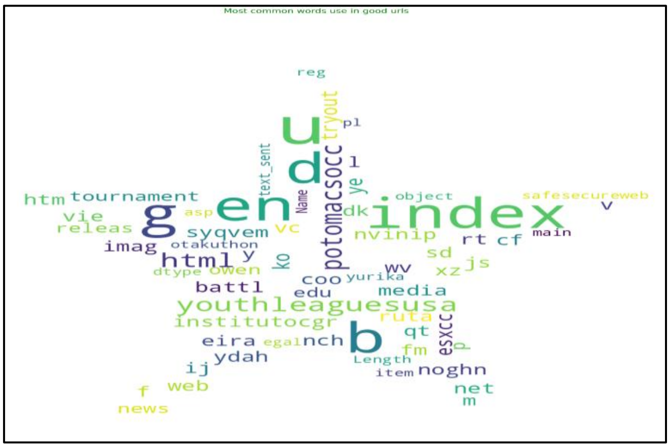

Figure 11: Word Cloud of good URLs

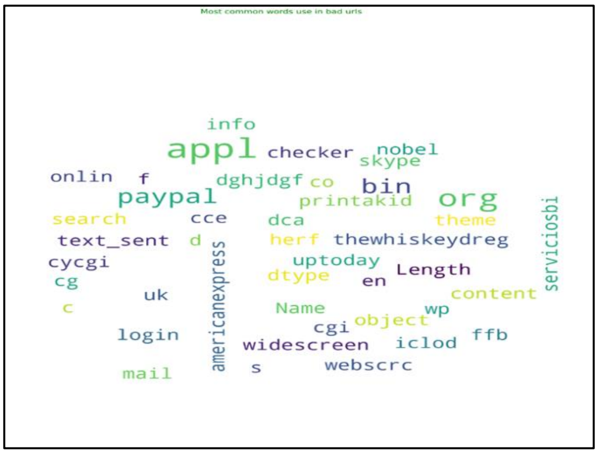

Figure 12: Word Cloud of bad URLs

CLASSIFICATION OF URL BASED ON LOGICAL REGRESSION (TRAINING)

The classification algorithm in the logistic regression model is to predict the chance of a classification rule. The dependent variable in logistic regression is a binary classification problem that contains data that is coded as 1 or 0. We train the model with some percentage of the dataset. For binary classification, this method is used. The connection between the predicting variable and at least one independent variable is evaluated using logistic regression (our features). To transform chances to binary values, the sigmoid function is applied. The Sigmoid-Function is a curve that acknowledges a real-valued input and restores values between 0 and 1, but never precisely at those limits. The values between 0 and 1 are again re-transformed to either 0 or 1 using a transformers classifier.

```
LogisticRegression
LogisticRegression()
```

PREDICTION OF URL BASED ON LOGISTIC REGRESSION (TESTING)

Once the model is trained, we test the model using the remaining percentage of the dataset for each algorithm. Based on the accuracy, we vary the testing percentage of the model.

- Accuracy

```
0.962665320704422
```

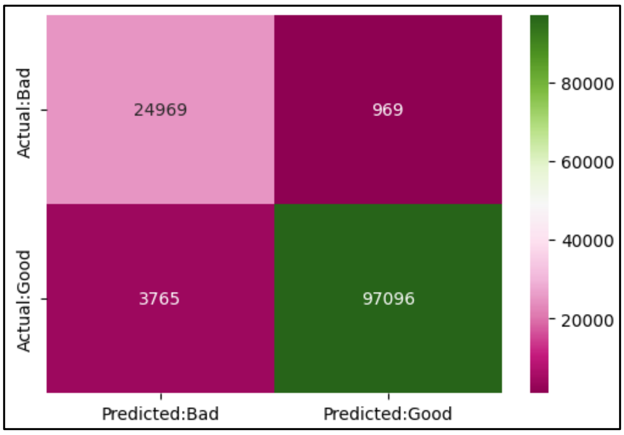

Figure 13: Confusion Matrix for Botnet Prevention

CONCLUSION

Threats to cyber security are changing, becoming much more undetected and complicated. Detecting harmful security risks and attacks is becoming a significant challenge in cyberspace. Machine learning is a powerful tool to overcome these challenges. The implementation of all the 6 modules has been achieved. The first part of the work involving ‘Classification and Prediction of DDoS attacks’ is with regards to the pre-processing of data. Using encoding techniques, conversion of categorical data to numerical data was performed. Next, normalization of the dataset using Feature Scaling and Feature Selection was performed and was optimized for the best features. After the preprocessing of the dataset was done, the model was trained using training dataset for 3 machine learning algorithms (Naive Bayes, Random Forest and XGBoost) and deep learning algorithm, Deep Neural Network. Once the model is trained, it was tested using the test data set and it gave good accuracies. The classification report was generated to analyse the performances of the various models created using different algorithms. By comparing the performances of all the four algorithms, it was found that Random Forest gave 99.36% accuracy, Naive Bayes gave 93.82% accuracy, XGBoost gave 98.925% accuracy and Deep Neural Network gave 90.17% accuracy. So, it can be concluded that Random Forest algorithm classifies the taken dataset more accurately. The next part of the work involving ‘Botnet Prevention’ majorly consists of tokenization. As a result of tokenization, the array of tokens was used to train the model using Logistic Regression algorithm. An accuracy of 97% was achieved for training set and 96% for testing set.

REFERENCES

[1] Jiang, H., He, Z., Ye, G., & Zhang, H. (2020). Network intrusion detection based on PSO-XGBoost model. IEEE Access, 8, 58392-58401

[2] JSu, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020). BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE Access, 8, 29575-29585

[3] Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An adaptive ensemble machine learning model for intrusion detection. IEEE Access, 7, 82512- 82521

[4] D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2019). Classification hardness for supervised learners on 20 years of intrusion detection data. IEEE Access, 7, 167455-167469

[5] Liu, C., Liu, Y., Yan, Y., & Wang, J. (2020). An intrusion detection model with hierarchical attention mechanism. IEEE Access, 8, 67542-67554

[6] Zhao, R., Mu, Y., Zou, L., & Wen, X. (2022). A hybrid intrusion detection system based on feature selection and weighted stacking classifier. IEEE Access, 10, 71414-71426

[7] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345-357

[8] Long, C., Xiao, X., Wan, W., Zhao, J., Wei, J., & Du, G. (2021, June). Botnet Detection Based on Flow Summary and Graph Sampling with Machine Learning. In 2021 International Conference on Computer Engineering and Application (ICCEA) (pp. 309-317). IEEE

[9] Mohmand, M. I., Hussain, H., Khan, A. A., Ullah, U., Zakarya, M., Ahmed, A.,... & Haleem,M. (2022). A machine learning-based classification and prediction technique for DDoS attacks.IEEE Access, 10, 21443-21454

[10] Chavan, N., Kukreja, M., Jagwani, G., Nishad, N., & Deb, N. (2022, March), DDoS Attack Detection and Botnet Prevention using Machine Learning. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 1159-1163). IEEE