International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

ENHANCED DDOS DETECTION USING MACHINE LEARNING

P. SIVA PRASAD, Assistant Professor, Department of MCA, Chirala engineering college, Chirala, Lakshmiprasad8216@gmail.com

RAVELABHANU PRAKASH, PG Student-MCA, Department of MCA, Chirala engineering college, Chirala, Ravelabhanuprakash@gamil.com

Abstract: The project aims to develop a system for detecting Distributed Denial of Service (DDoS) attacks using advanced techniques, specifically machine learning algorithms like Logistic Regression, K-Nearest Neighbor, and Random Forest. The proposed system is designed to achieve a high level of accuracy in detecting DDoS attacks. This is a crucial aspect of effectively mitigating such attacks, as accurate detection is key. The project seeks to go beyond existing methods and significantly enhance the capabilities of DDoS attack detection. This indicates a focus on innovation and better solutions. To assess the proposed models, the project opts for the NSL KDD dataset. This dataset is chosen because it is less redundant compared to others, resulting in better, more accurate results during testing and evaluation. The project conducts a comparative study involving various machine learning classifiers, including Logistic Regression, Random Forests, Decision Tree, and K-Nearest Neighbor. This comparison helps showcase the strengths and weaknesses of different algorithms in the context of DDoS attack detection. The project methodology involves a structured process, including data acquisition, feature extraction, and classification. It utilizes network properties and behaviors as essential features for the detection process. This signifies a systematic approach to building the DDoS detection system.And also included the DDoS detection project, a robust ensemble approach has been implemented with a Voting Classifier, combining Random Forest and AdaBoost, and a Stacking Classifier, merging Random Forest, MLP with LightGBM. Through this aim to further elevate the system's performance by leveraging the complementary strengths of diverse machine learning algorithms.

***Index terms** -DDoS; Deep Learning; Random Forest; Logistic Regression; KNN; NSL KDD Dataset.*

1. INTRODUCTION

The constant increase of around 5.03 billion internet users across the world widens the avenue of serious threat to the cyber security [1]. According to the study, in the third quarter of 2022, there was a 90% increase in the number of Distributed Denial-of-Service (DDoS) attacks compared to the same period in the previous year. [2]. The DDoS attacks impacts heavy losses to the infrastructure, industry, government and economy which is evident from the report in [3]. The DDoS attacks are a unique type of attempt where the online services of a specific web server are disrupted with malign intent.

A variant of cyber-attack identified as a distributed denial of service (DDoS) attack includes a large number of compromised devices, frequently dispersed globally in a botnet, to bombard a target with malicious traffic. Contrast this with a DoS assault, which employs a single device to bombard a target with traffic. Volume-based attacks, Protocol-based attacks, and Application layer attacks are the three subtypes of DDoS attacks [16]. Volume-based attacks, including UDP floods and ICMP floods, are measured in bits per second and attempt to saturate the target's bandwidth (Bps). SYN floods and Ping of Death are types of protocol-based attacks that can drain the resources of servers, firewalls, and load balancers by sending a large number of packets per second (Pps). These attacks can cause communication tools to become overwhelmed and unavailable. Attacks on the application layer, such as GET/POST floods and low-and-slow attacks, are calculated in requests per second and are intended to bring down the web server (Rps) [4]. These attacks are often disguised as legitimate and innocent requests.

DDoS attacks have been successful in interrupting crucial Internet system like DNS and costing businesses that depend on these services revenue. A majority (over 65%) of DDoS attacks are volumetric, which means that they involve the delivery of a large volume of irrelevant data with the aim of overwhelming the victim's computational resources or the capacity of nearby network links. The usage of common mechanism in queuing rules like First in First out and DropTail in internet routers makes these attacks effective that do not differentiate between different types of traffic and handle all traffic equally, including both attack and legitimate traffic.. As a result, these attacks can be successful in impairing the victim's ability to handle incoming data. Lowvolume DDoS attacks [2, 8, 10], on the other hand, are stealthier and typically exploit application layer protocols to exhaust victim resources without overloading links. These attacks can last only a few minutes or less than an hour, making them difficult to detect with common tools.

Based on our findings, we propose a deep learning method for detecting Distributed Denial of Service (DDoS) attacks [16] that involves data acquisition, feature extraction and classification, and binary classification. The proposed method utilizes network properties such as packet length, inter-packet intervals, and protocol, as well as network behaviors as features. We evaluate the performance of various attack detection classifiers, including Logistic Regression, Random Forests, Decision Tree, and K-Nearest Neighbor. To validate our proposed method, we use the NSL KDD dataset in our experiments, and the results obtained from this study are highly promising.

2. LITERATURE SURVEY

Smart farming also known as precision agriculture is gaining more traction for its promising potential to fulfill increasing global food demand and supply. In a smart farm, technologies and connected devices are used in a variety of ways, from finding the real-time status of crops and soil moisture content to deploying drones to assist with tasks such as applying pesticide spray. However, the use of heterogeneous internet-connected devices has introduced numerous vulnerabilities within the smart farm ecosystem. Attackers can exploit these vulnerabilities to remotely control and disrupt data flowing from/to on-field sensors and autonomous vehicles like smart tractors and drones. This can cause devastating consequences especially during a high-risk time, such as harvesting, where live-monitoring is critical. In this paper [3], we demonstrate a Denial of Service (DoS) attack that can hinder the functionality of a smart farm by disrupting deployed on-field sensors [9, 12, 16]. In particular, we discuss a Wi-Fi deauthentication attack that exploits IEEE 802.11 vulnerabilities, where the management frames are not encrypted. A MakerFocus ESP8266 Development Board WiFiDeauther Monster is used to detach the connected Raspberry Pi from the network and prevent sensor data from being sent to the remote cloud. Additionally, this attack was expanded to include the entire network, obstructing all devices from connecting to the network. To this end, we urge practitioners to be aware of current vulnerabilities when deploying smart farming ecosystems and encourage the cyber-security community to further investigate the domain-specific characteristics of smart farming.

In the current world, the Internet is being used almost everywhere. With the rise of IoT technology, which is one of the most used technologies, billions of IoT devices [5, 7]are interconnected over the Internet. However, DoS/DDoS attacks are the most frequent and perilous threat to this growing technology. New types of DDoS attacks are highly advanced and complicated, and it is almost impossible to detect or mitigate by the existing intrusion detection systems and traditional methods. Fortunately, Big Data, Data mining, and Machine Learning technologies make it possible to detect DDoS traffic effectively. This paper [4] suggests a DDoS detection model based on data mining and machine learning techniques. For writing this paper, the latest available Dataset, CICDDoS2019, experimented with the most popular machine learning algorithms and specified the most correlated features with predicted classes are being used. It is discovered that AdaBoost and XGBoost were extraordinarily accurate and correctly predicted the type of network traffic with 100% accuracy. Future research can be extended by enhancing the model for multiclassification of different DDoS attack types and testing hybrid algorithms and newer datasets on this model.

Vulnerable IoT devices [13] are powerful platforms for building botnets that cause billion-dollar losses every year. In this work, we study Bashlite botnets and their successors, Mirai botnets. In particular, [5] we focus on the evolution of the malware as well as changes in botnet operator behavior. We use monitoring logs from 47 honeypots collected over 11 months. Our results shed new light on those botnets, and complement previous findings by providing evidence that malware, botnet operators, and malicious activity are becoming more sophisticated. Compared to its predecessor, we find Mirai uses more resilient hosting and control infrastructures, and supports more effective attacks.

Defending against distributed denial of service (DDoS) attacks in the Internet is a fundamental problem. One practical approach to addressing [7] DDoS attacks is to redirect all destination (e.g., via DNS or BGP) to a third-party, DDoS protection-as-a-service provider (e.g., Cloudflare and Akamai) that is well provisioned and equipped with proprietary filtering mechanisms to remove attack traffic before passing the remaining traffic to the destination. Although such an approach is appealing, as it requires no modification to the existing Internet infrastructure and can scale to handle very large attacks, recent industrial interviews with more than 100 interviewees from over 10 industry segments reveal that this approach alone is not sufficient, especially for large organizations (e.g., web hosting companies, government) that cannot afford to allow third-parity security-service providers to terminate their network connections. Instead, these organizations have to rely on their ISPs to filter attack traffic. In this paper, we discuss the challenges faced by the ISPs in order to disrupt the Internet security-service market and sketch our solutions, powered by smart contracts.

Currently, Distributed Denial of Service Attacks [2, 4, 6, 7, 8] are the most dangerous cyber danger. By inhibiting the server's ability to provide resources to genuine customers, the affected server's resources, such as bandwidth and buffer size, are slowed down. A mathematical model for distributed denial-of-service attacks is proposed in this study [9]. Machine learning algorithms such as Logistic Regression and Naive Bayes, are used to detect attacks and normal scenarios. The CAIDA 2007 Dataset is used for experimental study. The machine learning algorithms are trained and tested using this dataset and the trained algorithms are validated. Weka data mining platform are used in this study for implementation and results of the same are analysed and compared. Other machine learning algorithms used with respect to denial of service attacks are compared with the existing work.

Distributed Denial of Service attack (DDoS) [10, 11, 13] is the most dangerous attack in the field of network security. DDoS attack halts normal functionality of critical services of various online applications. Systems under DDoS attacks remain busy with false requests (Bots) rather than providing services to legitimate users. These attacks are increasing day by day and have become more and more sophisticated. So, it has become difficult to detect these attacks and secure online services from these attacks. In this paper [10], we have used machine learning [4, 9, 10] based approach to detect and classify different types of network traffic flows. The proposed approach is validated using a new dataset which is having mixture of various modern types of attacks such as HTTP flood, SID DoS and normal traffic. A machine learning tool called WEKA is used to classify various types of attacks. It has been observed that J48 algorithm produced best results as compared to Random Forest and Naïve Bayes algorithms.

3. METHODOLOGY

i) Proposed Work:

The proposed system uses machine learning algorithms, including Logistic Regression, K-Nearest Neighbor, and Random Forest, to enhance DDoS attack detection. It follows three key steps: data collection, feature extraction, and classification. This approach improves cybersecurity by accurately identifying DDoS attacks [1, 2, 3] in network traffic.In extending the DDoS detection project, a robust ensemble approach has been implemented with a Voting Classifier, combining Random Forest and AdaBoost, and a Stacking Classifier, merging Random Forest, MLP with LightGBM. These modelsaim to further elevate the system's performance by leveraging the complementary strengths of diverse machine learning algorithms [13]. Additionally, a user-friendly Flask framework integrated with SQLite has been developed, featuring signup and signin functionalities for effective user testing, enhancing the system's accessibility and practicality in real-world cybersecurity applications.

ii) System Architecture:

The system begins with the collection and preparation of a dataset. In this case, the NSL KDD dataset [8] is often used, which contains network traffic data with labeled instances of normal and DDoS attack traffic. Data preprocessing is a crucial step where raw data is cleaned, transformed, and made ready for analysis. This includes handling missing values, removing duplicates, and normalizing or scaling features to ensure consistency. Feature selection involves choosing the most relevant attributes or features from the dataset. This step aims to reduce dimensionality and improve the efficiency of the detection process. Common techniques include correlation analysis, mutual information, or feature importance ranking. This stage is the core of the system where various machine learning classification algorithms are employed to detect DDoS attacks. The mentioned algorithms - K-Nearest Neighbor (KNN), Logistic Regression (LR), and Random Forest (RF), EXTENSION VOTING CLASSIFIER AND STACKING CLASSIFIER- are utilized here. Each algorithm processes the preprocessed data to classify network traffic into normal or DDoS attack categories. To evaluate the effectiveness of the DDoS detection system, performance criteria are defined. These criteria include metrics such as accuracy, precision, recall, F1-score. These measures assess how well the system distinguishes between normal and attack traffic.

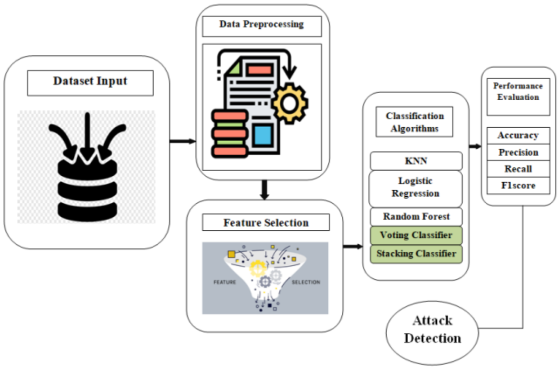


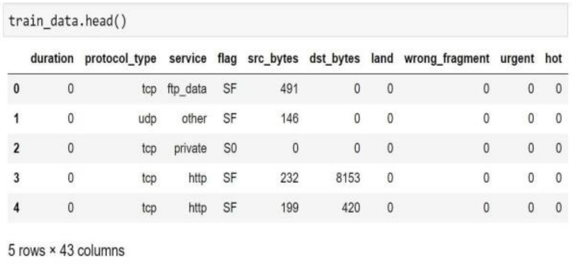
Fig 1 Proposed architecture

iii) Dataset collection:

NSL-KDD DATASET

We have used NSL-KDD dataset [8] in this project to train ml models.. It provides a diverse range of network traffic data that includes multiple attack types and enhanced labeling. This dataset serves as a valuable asset for evaluating intrusion detection systems and testing machine learning models within a well-balanced and controlled environment.

So, these are the top 5 rows of the nsl-kdd data set. So, it contains 43 columns, we are displaying few of them here.



```
train_data.head()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	0	tcp	ftp_data	SF	491	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0

5 rows × 43 columns

Fig 2 NSL KDD dataset

iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

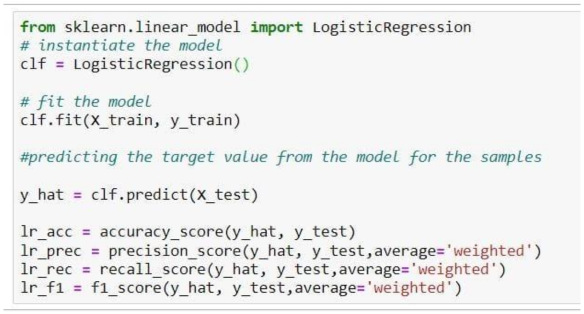
v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important [16].

vi) Algorithms:

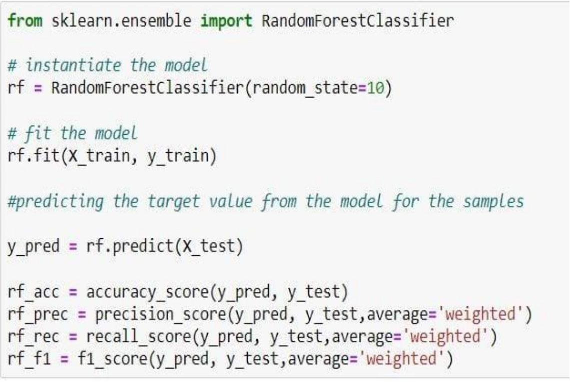
Logistic Regression- Logistic Regression is a classification technique used to predict the likelihood of an input falling into a particular category. It uses the sigmoid function to transform input features into a probability score, ranging from 0 to 1. By applying a threshold, the input is classified into one of multiple categories based on this probability. During training, the model adjusts its coefficients to optimize data fit and improve classification accuracy.



```
from sklearn.linear_model import LogisticRegression
# instantiate the model
clf = LogisticRegression()

# fit the model
clf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_hat = clf.predict(X_test)

lr_acc = accuracy_score(y_hat, y_test)
lr_prec = precision_score(y_hat, y_test,average='weighted')
lr_rec = recall_score(y_hat, y_test,average='weighted')
lr_f1 = f1_score(y_hat, y_test,average='weighted')
```

Fig 3 Logistic regression

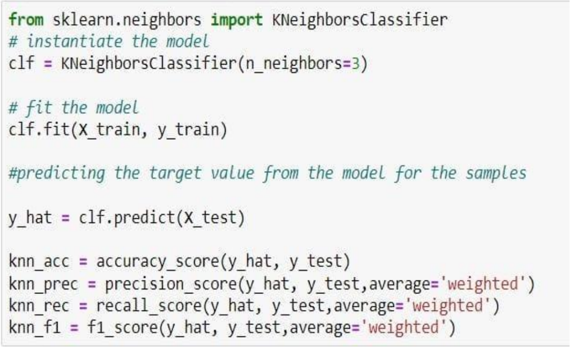
Random Forest: Random Forest is an ensemble learning technique that harnesses the collective power of multiple decision trees to generate predictions. It accomplishes this by training a group of decision trees on random portions of the data and subsequently aggregating their predictions. This ensemble strategy elevates prediction accuracy, mitigates overfitting issues, and delivers robust performance, suitable for tasks involving both classification and regression [7].



```
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
rf = RandomForestClassifier(random_state=10)

# fit the model
rf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = rf.predict(X_test)

rf_acc = accuracy_score(y_pred, y_test)
rf_prec = precision_score(y_pred, y_test,average='weighted')
rf_rec = recall_score(y_pred, y_test,average='weighted')
rf_f1 = f1_score(y_pred, y_test,average='weighted')
```

Fig 4 Random forest

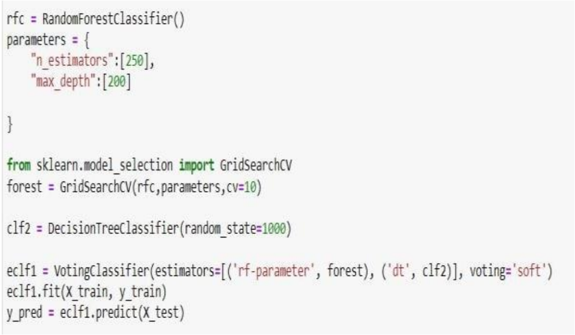
KNN : K-Nearest Neighbors (KNN) is a versatile machine learning algorithm used for classification and regression tasks. In the training phase, KNN stores data points and their corresponding labels or values. When predicting the class or value of a new data point, KNN identifies the K-nearest neighbors from the training set based on a chosen distance metric. For classification, it assigns the most common class label among these neighbors, while for regression, it calculates the average of their values. KNN relies on the principle that similar data points tend to share characteristics, making it an intuitive and easy-to-implement algorithm in various applications [7].



```
from sklearn.neighbors import KNeighborsClassifier
# instantiate the model
clf = KNeighborsClassifier(n_neighbors=3)

# fit the model
clf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_hat = clf.predict(X_test)

knn_acc = accuracy_score(y_hat, y_test)
knn_prec = precision_score(y_hat, y_test,average='weighted')
knn_rec = recall_score(y_hat, y_test,average='weighted')
knn_f1 = f1_score(y_hat, y_test,average='weighted')
```

Fig 5 KNN

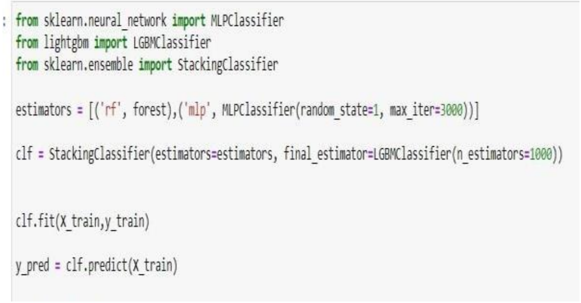
Voting Classifier : The Voting Classifier is an ensemble machine learning technique that combines the predictions of multiple base classifiers, like Random Forest and AdaBoost, to achieve higher accuracy. Random Forest uses multiple decision trees to mitigate overfitting, while AdaBoost trains weak learners in sequence, focusing on correcting misclassified samples. By consolidating these individual predictions through majority or weighted voting, the Voting Classifier leverages the diversity of models to enhance accuracy by rectifying errors made by each classifier, resulting in more robust predictions [7].



```
rfc = RandomForestClassifier()
parameters = {
    "n_estimators":[250],
    "max_depth":[200]

}

from sklearn.model_selection import GridSearchCV
forest = GridSearchCV(rfc,parameters,cv=10)

clf2 = DecisionTreeClassifier(random_state=1000)

eclf1 = VotingClassifier(estimators=[('rf-parameter', forest), ('dt', clf2)], voting='soft')
eclf1.fit(X_train, y_train)
y_pred = eclf1.predict(X_test)
```

Fig 6 Voting classifier

Stacking Classifier: The Stacking Classifier is an ensemble method that brings together three base models: Random Forest (RF), Multi-Layer Perceptron (MLP), and LightGBM. RF is known for its decision tree ensembles, MLP is a neural network capable of learning intricate patterns, and LightGBM is a gradient boosting framework. Stacking takes the predictions from these base models and uses them as input to train a meta-model. The goal is to boost prediction accuracy by capitalizing on the unique strengths of each base model and capturing complex data relationships, ultimately producing more precise predictions [7].



```
: from sklearn.neural_network import MLPClassifier
  from lightgbm import LGBMClassifier
  from sklearn.ensemble import StackingClassifier

  estimators = [('rf', forest),('mlp', MLPClassifier(random_state=1, max_iter=3000))]

  clf = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier(n_estimators=1000))


  clf.fit(X_train,y_train)

  y_pred = clf.predict(X_train)
```

Fig 7 Stacking classifier

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive}+\textit{False Positive}}$$

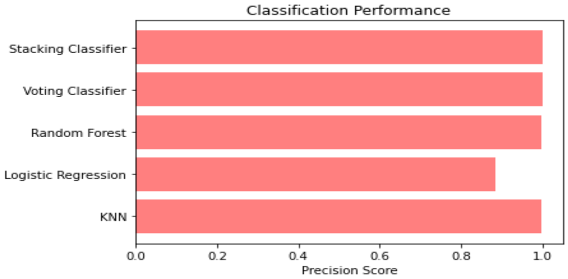


Fig 8 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

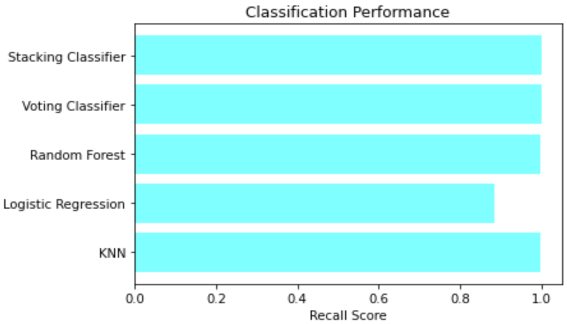


Fig 9 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

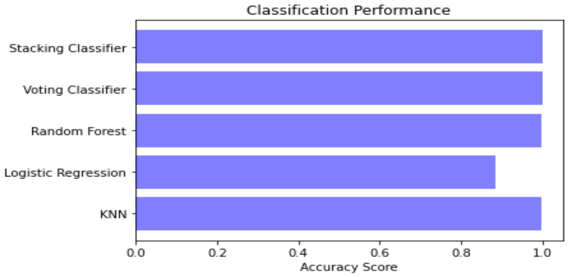


Fig 10 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

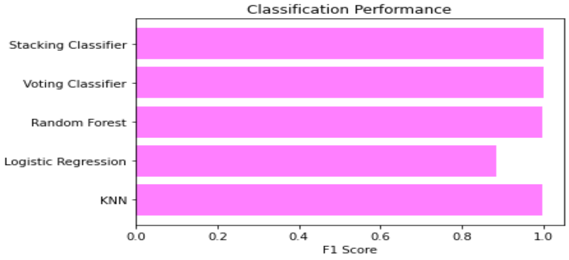
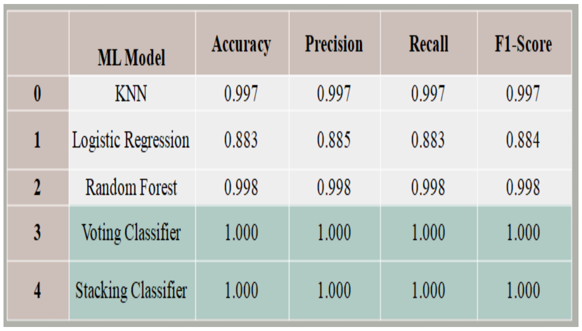
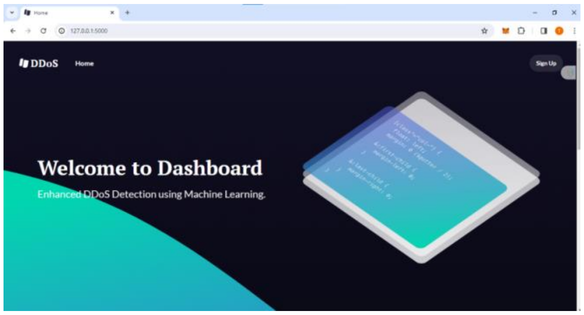


Fig 11 F1Score



	ML Model	Accuracy	Precision	Recall	F1-Score
0	KNN	0.997	0.997	0.997	0.997
1	Logistic Regression	0.883	0.885	0.883	0.884
2	Random Forest	0.998	0.998	0.998	0.998
3	Voting Classifier	1.000	1.000	1.000	1.000
4	Stacking Classifier	1.000	1.000	1.000	1.000

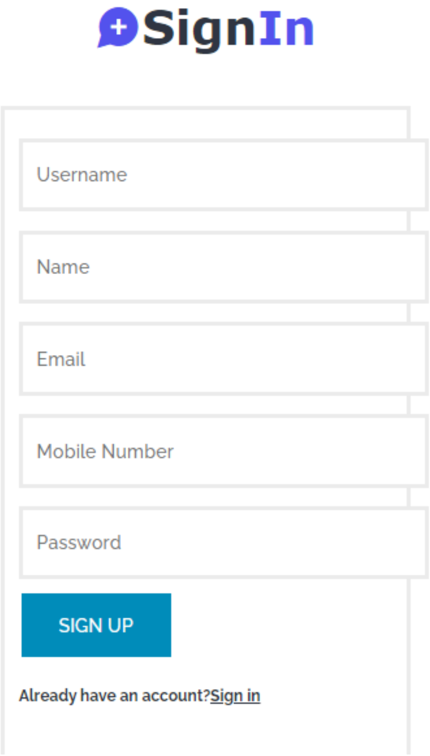
Fig 12 Performance Evaluation



Welcome to Dashboard

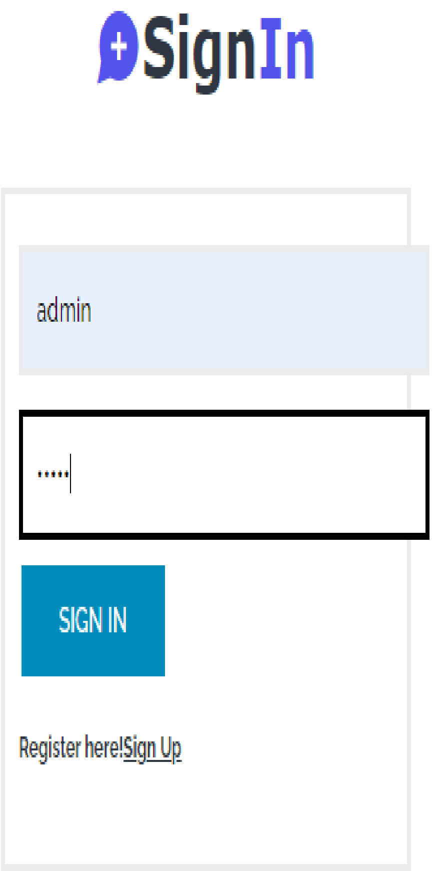
Enhanced DDoS Detection using Machine Learning.

Fig 13 Home page



Already have an account?Sign in

Fig 14 Signin page



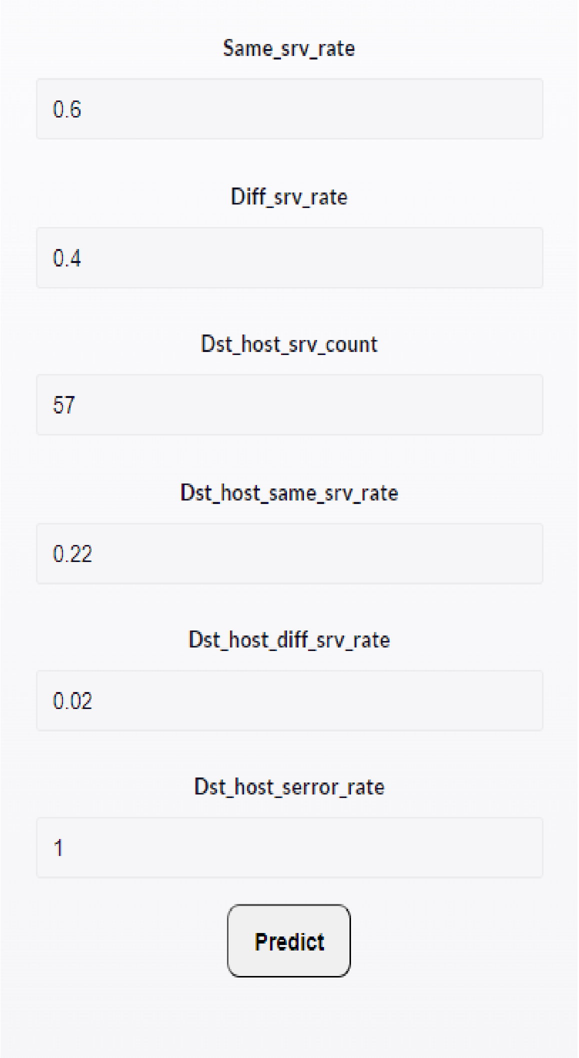
SignIn

admin

•••••

Register here!Sign Up

Fig 15 Login page



Same_srv_rate

0.6

Diff_srv_rate

0.4

Dst_host_srv_count

57

Dst_host_same_srv_rate

0.22

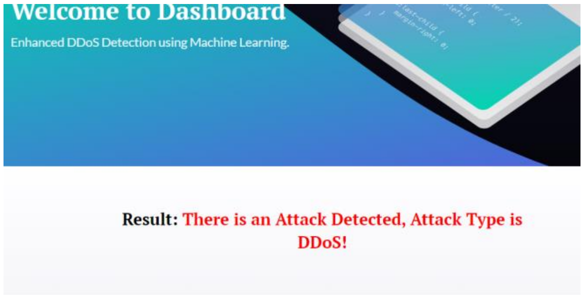
Dst_host_diff_srv_rate

0.02

Dst_host_serror_rate

1

Fig 16User input



Result: There is an Attack Detected, Attack Type is DDoS!

Fig 17 Predict result for given input

5. CONCLUSION

The project makes digital systems safer by better detecting DDoS attacks, ensuring that online services stay up and digital assets are protected. We use different machine learning models like K-Nearest Neighbors (KNN), Logistic Regression, and Random Forest to find and respond to DDoS attacks effectively [7]. We improve accuracy a lot by using special techniques as an extension, like Voting Classifier and Stacking Classifier. They combine the predictions of multiple models to make our system stronger and more reliable. Going beyond algorithm development, the project integrates a user-friendly front end using the Flask framework. This practical implementation facilitates user testing, enabling stakeholders to interact seamlessly with the model. The incorporation of user authentication with SQLite ensures secure sign-up and sign-in processes, elevating the overall user experience. This project shows how machine learning can enhance security by detecting DDoS attacks effectively [2, 3]. Ensemble models, especially, have proven to be very accurate and dependable. Many people benefit from this project, including organizations, network admins, cybersecurity experts, and end users. It makes online services more reliable and secure for everyone.

6. FUTURE SCOPE

In the future, we can make the system better by adding real-time DDoS detection tools, which will make it faster and more effective [15]. We can also try different ways to pick out the most important factors for predicting DDoS attacks more accurately. By testing more machine learning methods, we can expand the system's abilities beyond what it can do with the current methods like Logistic Regression, Random Forests, and K-Nearest Neighbor [13]. We should also work on making the system able to find smaller DDoS attacks that use tricky methods, which are hard to spot right now. And finally, we can keep improving the system's accuracy and how well it works by emplying deep learning and using better ways to analyze network traffic.

REFERENCES

[1] Statista Research Department, “Worldwide digital population July 2022”, Available: https://www.statista.com/statistics/617136/digitalpopulation-worldwide/ (Last Accessed on: December 31, 2022)

[2] Ramil Khantimirov, "DDoS Attacks in 2022: Trends and Obstacles Amid Worldwide Political Crisis", Available: https://www.infosecurity-magazine.com/blogs/ddos-attacks-in-2022- trends/ (Last Accessed on: December 31, 2022)

[3] S. Sontowski et al., "Cyber Attacks on Smart Farming Infrastructure," 2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC), 2020, pp. 135-143, doi: 10.1109/CIC50333.2020.00025.

[4] Seifousadati, Alireza and Ghasemshirazi, Saeid and Fathian, Mohammad, "A Machine Learning Approach for DDoS Detection on IoT Devices", arXiv, 2021. Doi: 10.48550/ARXIV.2110.14911

[5] A. Marzano, D. Alexander, O. Fonseca et al., "The evolution of bashlite and mirai IoT botnets," in Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), 2018.

[6] S. Kottler, "February 28th DDoS incident report," 2018, https://github.blog/2018-03-01-ddos-incident-report/.

[7] Y. Cao, Y. Gao, R. Tan, Q. Han, and Z. Liu, "Understanding internet DDoS mitigation from academic and industrial perspectives," IEEE Access, vol. 6, pp. 66641–66648, 2018.

[8] S. Newman, "Under the radar: the danger of stealthy DDoS attacks," Network Security, vol. 2019, no. 2, pp. 18-19, 2019.

[9] Kumari, K., Mrunalini, M., "Detecting Denial of Service attacks using machine learning algorithms", . J Big Data 9, 56 (2022).

[10] P. S. Saini, S. Behal and S. Bhatia, "Detection of DDoS Attacks using Machine Learning Algorithms," 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), 2020, pp. 16-21, doi: 10.23919/INDIACom49435.2020.9083716.

[11] Jiangtao Pei et al " A DDoS Detection Method based on Machine Learning", J. Phys.: Conf. Ser. 1237 032040, 2019.

[12] Abdullah Soliman Alshra'a, Ahmad Farhat, Jochen Seitz, "Deep Learning Algorithms for Detecting Denial of Service Attacks in Software-Defined Networks", Procedia Computer Science, Volume 191, 2021, Pages 254-263, ISSN 1877-0509.

[13] Seifousadati, Alireza, Saeid Ghasemshirazi, and Mohammad Fathian. "A Machine Learning Approach for DDoS Detection on IoT Devices." arXiv preprintr Xiv:2110.14911 (2021).

[14] Francisco Sales de Lima Filho, Frederico A. F. Silveira, Agostinho de Medeiros Brito Junior, Genoveva Vargas-Solar, Luiz F. Silveira, "Smart

Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning", Security and Communication Networks, vol. 2019, Article ID 1574749, 15 pages, 2019.

[15] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 29-35, doi: 10.1109/SPW.2018.00013.

[16] Ebtihal Sameer Alghoson, Onytra Abbass, “Detecting Distributed Denial of Service Attacks using Machine Learning Models”, International Journal of Advanced Computer Science and Applications, Vol. 12, No. 12, 2021.