Dynamic Web Application for Real-Time threat Detection using ML Algorithms

Dr.M.Dhanalakshmi[1], Kapilavai Pravallika[2]

[1] *Professor of IT, University College of Engineering, Science and Technology Hyderabad, JNTUH, Hyderabad, India*

[2] *Masters Student of CNIS, University College of Engineering, Science and Technology Hyderabad, JNTUH, Hyderabad, India*

***Abstract*— In today's highly interconnected world, the security of computer networks is paramount. Conventional systems have difficulty identifying new attacks and frequently produce a significant number of false positives. In this project the real-time threat detection and classification is done by using Categorical Boosting (CatBoost), k-medoids clustering and Deep Q Networking (DQN) algorithms. Later, dynamic web application is build on top of CatBoost algorithm that helps the user to enter a string value that has the details about the network packet and help to classify it as normal or malicious attack type. The time complexity for the algorithms is comparatively higher than other supervised algorithms but these algorithms are good for outlier analysis. The CatBoost algorithm gave an accuracy of 99.24% for the multi-class classification and other algorithms gave a decent accuracy score when combining them with supervised model for the binary and multi-class classification.**

***Index Terms*— Network Intrusions, Network intrusion detection, Categorical Boosting, K-Medoids, Deep Q Network, Streamlit API, Normal, DoS, R2L, U2R, Probe, Anomaly.**

I. INTRODUCTION

As cyber threats become more sophisticated and pervasive, traditional network intrusion detection systems (NIDS) that rely on static, signature-based methods are increasingly inadequate. Machine learning (ML) offers a promising solution to these challenges by enabling NIDS to learn and adapt from data, thereby improving their ability to detect both known and unknown attacks. Unlike signature-based approaches, ML-based intrusion detection systems (IDS) analyze patterns and behaviors within network traffic to identify anomalies that may signify malicious activity. By leveraging large datasets and advanced algorithms, ML can discern subtle deviations from normal behavior, enhancing the detection accuracy and reducing the incidence of false alarms.

In this project, the partial version of the [1]NSL-KDD dataset is used for the classification and detection of the malicious and normal attack classes. The [2] NSL-KDD dataset contains 42 features which help in understand the network connection data. Supervised learning algorithm that is categorical boosting (CatBoost), unsupervised learning algorithm that is k medoids clustering algorithm, and deep reinforcement learning algorithm that is deep Q network (DQN) are implemented and compared for binary and multiple class classification and also for the outlier performance improvement for the real-world network connection analysis.

II. LITERATURE SURVEY

The existing works [3], [4], [5] depicts the comparison of the machine learning models for the NIDS. These are mostly preformed on diverse datasets including KDD Cup 99, NSLKDD /CIC-IDS 2017, UNSW, NB-15, and UGR ’16. The binary and multi class classification is performed on these dataset with supervised learning models. [6] X. Zhang and et al. introduced a hybrid model which is the combination of the Convolutional Neural Network (CNN) and Synthetic Minority Oversampling Technique combined with Edited Nearest Neighbors (SMOTE-ENN) algorithm. This model achieved an accuracy of 83.31% for network data. This method is useful for imbalance network traffic but is complex for normal network traffic. [7]

Jain and et al. proposed a CatBoost model which gave 70.79% and 87.65% accuracy for the multi-class and binary class classification respectively. [8] Hsu and et al. proposed a deep learning model which is performed on different datasets which gave an accuracy of around 92% average. [9] Rajat Bajwa and et al. proposed improved k-medoids clustering algorithm which is compare with k-mean and k-medoids algorithm. This model gave an accuracy of 99%. [10] Bahjat and et al. introduced hierarchical classification and clustering methods in combination with Sequential Forward Feature Selection (SFFS) feature selection for network intrusion detection. The accuracy of this model is done for individual categories i.e., normal and attack classes which ranged from 86% to 99%.

The existing machine leaning work on network intrusion detection gave a significant analysis of understanding, detecting and classifying the normal and anomaly classes. The existing work includes binary as well as multi-class classification and detection using wide range of machine learning algorithms. The most common disadvantages of the existing work are high false positives for fresh network data and time complexity for computing the results. When the false positives of the models are low then the time complexity of the model is high. And when the time complexity is low then the false positives of the data are high. Which signify the major disadvantage of the existing models. The other disadvantages of the existing systems include high dimensionality of the network data and imbalanced data analysis that lead to the inconsistent results of the models.

III. METHODOLOGY

Firstly, the system architecture of the proposed system is as shown in the fig.1. The frontend and backend of the application are depicted where the predicted output is send to the user at the end. Fig.2 shows the machine learning (ML) model implementation for the chosen supervised, unsupervised and reinforcement learning models i.e., the CatBoost, k-medoids clustering, and DQN respectively.

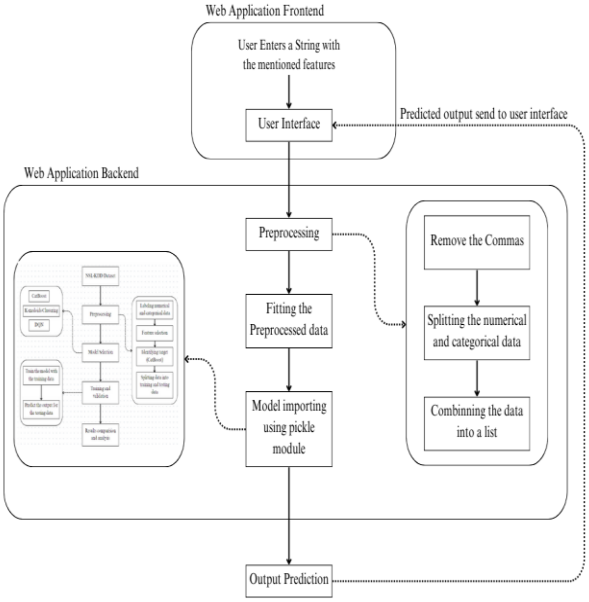

Web Application Frontend
User Enters a String with the mentioned features
User Interface
Predicted output send to user interface
Web Application Backend
Preprocessing
Remove the Commas
Splitting the numerical and categorical data
Combinning the data into a list
Fitting the Preprocessed data
Model importing using pickle module
Output Prediction

Fig.1 System Architecture of Web Application for real time threat detection

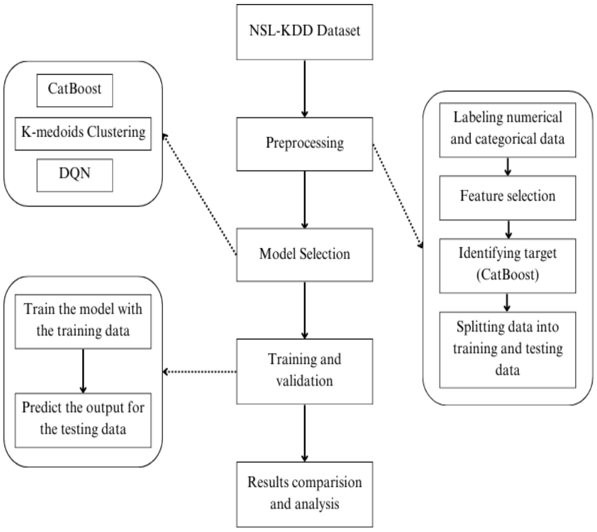

NSL-KDD Dataset
CatBoost
K-medoids Clustering
DQN
Preprocessing
Labeling numerical and categorical data
Feature selection
Identifying target (CatBoost)
Splitting data into training and testing data
Model Selection
Train the model with the training data
Predict the output for the testing data
Training and validation
Results comparision and analysis

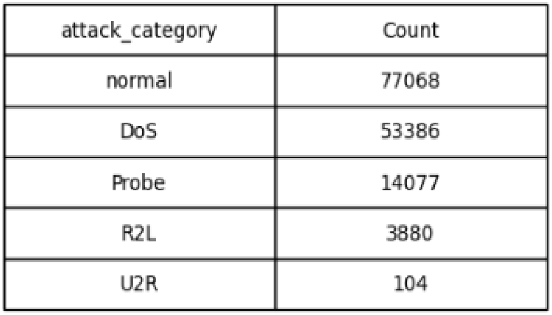
Fig.2 General ML Model Implementation for classification and detection

A. Dataset Description

Dataset used is the refined and reduced version of NSL-KDD dataset [1], [12], [13], [14]. Dataset contains network traffic data that has normal and attack connections. This data is widely used for more than two decades for research in machine learning in network intrusion detection. Table.1 shows the table of number of records corresponding to different class type. The dataset contains 41 features which are extracted from individual and multiple connections.

Every connection has a unique attack type assigned to it, and is classified as either normal or an attack.

Table.1: Dataset details with attack categories count



attack_category	Count
normal	77068
DoS	53386
Probe	14077
R2L	3880
U2R	104

DoS (Denial-of-service), R2L (Remote to Local), U2R (User to Root), and probing are the four primary types of attacks. The attack classes demonstrates the major attack groups that are

B. ML Model Implementation

The flowing algorithm flows are the implementation of the CatBoost, k-medoids clustering, and DQN algorithms [15]. Fig.3 Shows the DQN model work flow for the detection of anomaly and normal categories of the network data packets. For DQN algorithm the most relevant features are selected [11]. The k-medoids clustering and DQN model are later combined with the random forest algorithm for the binary classification.

Algorithm-1: CatBoost Algorithm for the implementation.

```
Step-1: Initializing the model:   $G_0(a)$ = Mean(b)
Step-2: Iteratively update the model:
        For n=1 to N:
            • Pseudo-residuals are computed:
                  $r_{im} = b_i – G_{m-1}(a_i)$
            • Fit a symmetric tree $g_m(a)$ to the pseudo-residuals.
            • Update the model:
                  $G_m(a) = G_{m-1}(a) + v g_m(a)$
```

Algorithm-2: K-medoids clustering algorithm for the implementation.

```
Step-1: Select k initial medoids randomly from the data points.
Step-2: Assign each data point to the nearest medoid:
        $C_i = \{ x_j : d(x_j,m_j) < d(x_j,m_h) \text{ for all } h \neq i \}$
Where, $d(x_j,m_j)$ is the dissimilarity between the point $x_j$ and $m_j$.
Step-3: Update the medoids: For each cluster $C_i$, find the new medoid $m_i$ that minimizes.
        $\sum_{xj\in Ci} d(xj,mi)$
Step-4: Repeat the assignment and update steps until the medoids do not change or until a specified number of iterations are reached.
Step-5: The objective function to minimize in k-medoids clustering is the sum of the pair wise dissimilarities between points and their medoids:
        $\sum^{k}_{i=1} \sum_{x\in Ci} d(x,mi)$
```

Algorithm-3: Deep Q Network algorithm flow for the implementation.

```
Step-1: Initialize the experimental replay buffer B, primary network T{s(t),a(t)} and target network T'{s(t),a(t)}.
Step-2: For each episode: set up the state $S_0$.
    • For probability ϵ select a random action a, otherwise select.
          $a=arg\{max_a T\{s(t),a(t)\}\}$
    • See the reward $r$ and next state $s(t+1)$ after executing action $a(t)$.
    • In replay buffer B store the transition {s(t), a(t), r, s(t+1)}.
    • A random mini-batch of transitions $\{s_j(t), a_j(t), r_j(t), s_j(t+1)\}$ from B are sampled.
    • The target is computed for every mini-batch transition.
    • Gradient decent is performed on the loss function.
    • Update the target network parameters regularly.
```

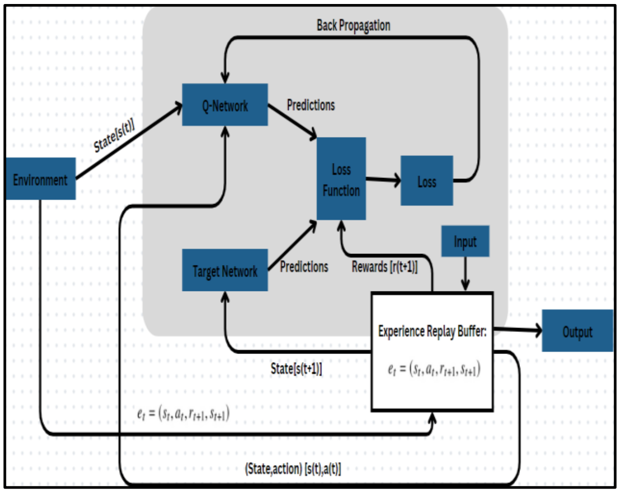


Fig.3 DQN algorithm work flow

C. Web Application Implementation

Streamlit API is used to build the web application that takes a string value with 41 features or with just the basic features of the network data packet which are separated by a comma. The web application has an user interface which contains the details about the project i.e., the accuracy score obtained by the model, features that need to be entered, text field with a check box that enables the button for the prediction and web application contains sidebar that contains basic features like resources, about the project and contact details. Following flow shows the detail outline of the web application implementation for the user data of network packets.

Flow: Web application constructed using streamlit API flow.

Enter a String: “String containing features” → S
Select the check box and click predict.
Suppose lists: A and B; Where A=Numerical list, B=Categorical list.
String ‘S’ is then split at comma.
Separation: Numerical and Categorical values into A and B list.
Combining to list: Cascading the A and B to form the desired list for prediction.

Output: Attack class; Description; Causes; Recommendation.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. CatBoost Algorithm Results

The performance analysis of the CatBoost algorithm [16] for multi-class classification is generated and a confusion matrix for all the classes is been produced. For this algorithm, it took around two and half hours to execute for the 1000 iterations of the different training data. The accuracy of the model is 99.2459% which is comparatively low with other supervised algorithms but, the outlier performance of this model is quite high than other algorithms. The fig.4 shows the confusion matrix which is plotted for predicted and actual classes of the records of the given data and fig.5 shows the analysis metric scores for train and test data.

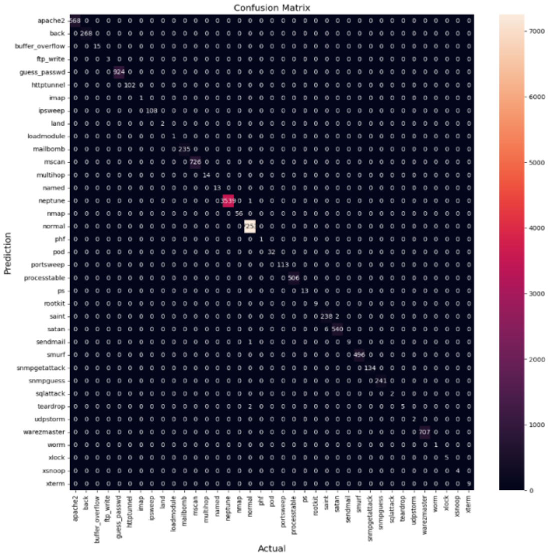


Fig. 4 Confusion matrix for CatBoost Classification

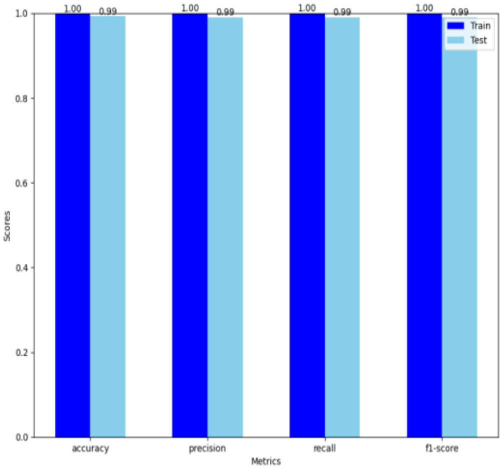


Fig.5 Analysis metrics Vs Scores of CatBoost algorithm

B. K-medoids Clustering Algorithm Results

For k-medoids clustering algorithms [17], [18] the performance analysis is done by using the silhoutte score. The sihoutte score values are shown in the fig.6 with respect to the number of clusters taken in the code. For the clusters (k = 3), the silhoutte score is average and is equal to silhoutte scores of clusters with high number. Fig.7 and fig.8 shows the 2-D plots that is t-SME and PCA graphs for the Clusters (k=3) are plotted respectively. It took around three hours to run and execute the algorithm. But it worked good for the outliers to make them fit in one of the clusters.

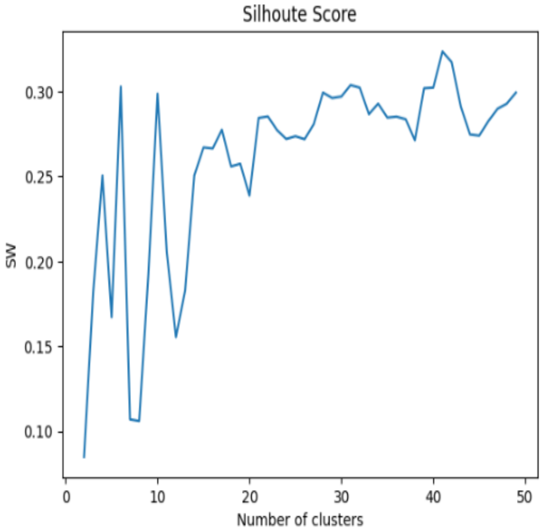


Fig.6 Silhoutte score Vs No. of clusters graph of k-medoids algorithm

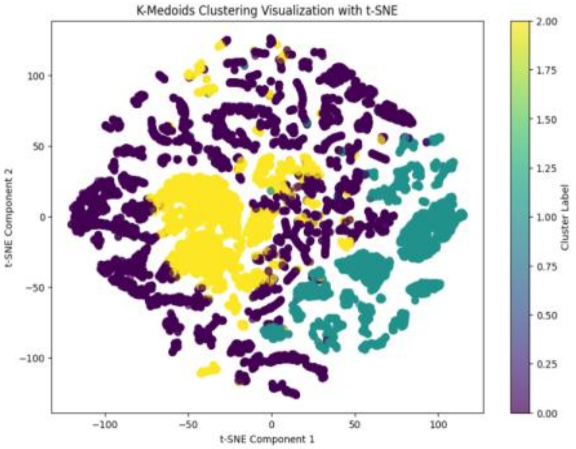


Fig.7 t-SNE plot for k=3 clusters of K-medoids clustering algorithm

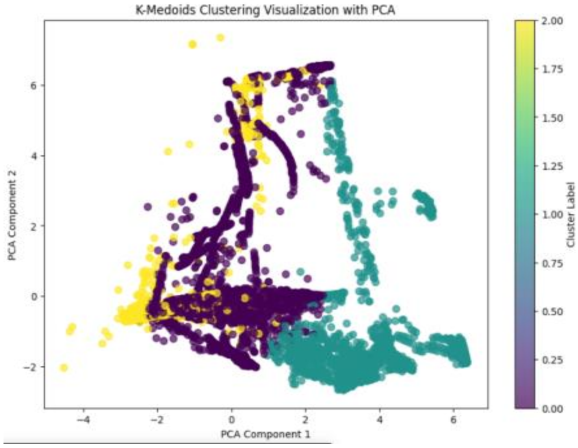


Fig.8 PCA plot for k=3 clusters of K-medoids clustering algorithm

The k-medoids clustering algorithm is then combined with the random forest classifier for binary classification of the records into attack and normal classes. This combined model gave an accuracy of 92.14% for the binary classification. The classification analysis is illustrated in the confusion matrix as shown in the fig.9. They show the actual and predicted labels for the normal and anomaly categories for the testing and training data.

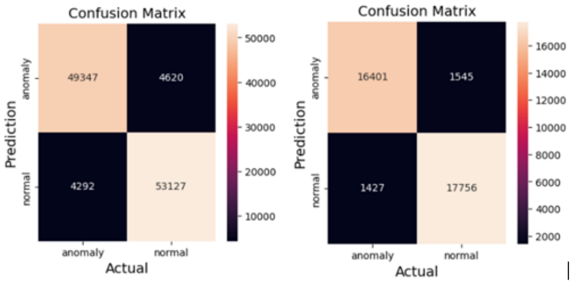


Fig.9 Confusion matrixes of train and test data of k-medoids algorithm

C. DQN Algorithm Results

Deep Q Network algorithm [19], [20] is a deep reinforcement learning algorithm that has the state, action, and reward as the three basic components. The DQN algorithm has gave a decent accuracy score for the detection of the class of the multi-class data. The testing evaluation results for this algorithm are accuracy is around 70%. Fig.10 shows the Loss Vs training steps graph, the loss rate has constantly decreased along the training and Fig.11 shows the

Reward Vs Episodes graph where the reward indicate the positive outcome or positive prediction and it increased gradually with respect to each episode. Fig.12 shows the entropy graph of the DQN model while training to predict the accurate results.

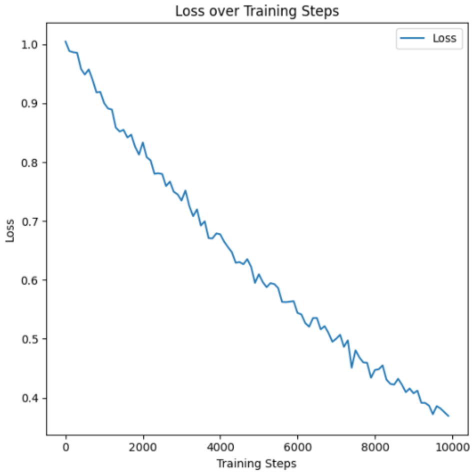


Fig.10 Loss Vs Training Steps graph of DQN model

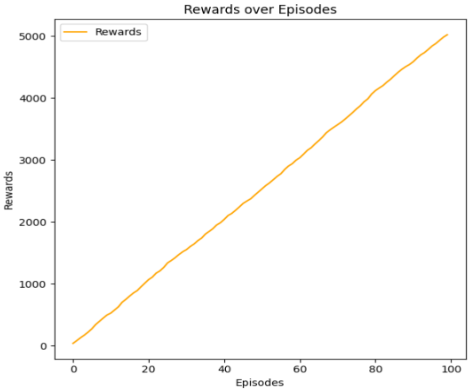


Fig.11 Rewards Vs Episodes graph of DQN model

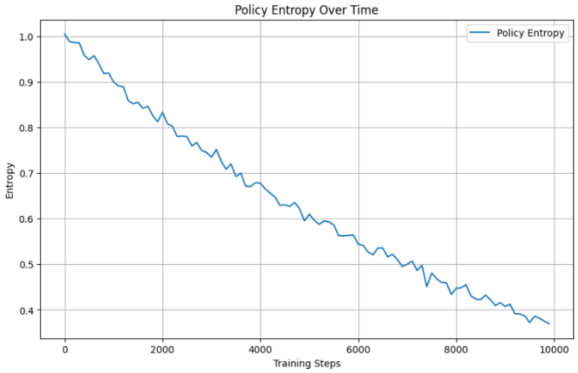


Fig.12 Entropy Vs Training steps graph of DQN model

The DQN model is then combined with the random forest algorithm for the binary classification of the respective data into the categories as normal and anomaly. The combined model gave accuracy approximately equal to 85.89% for the binary classification. This classification is illustrated in the fig.13 for the train and test data.

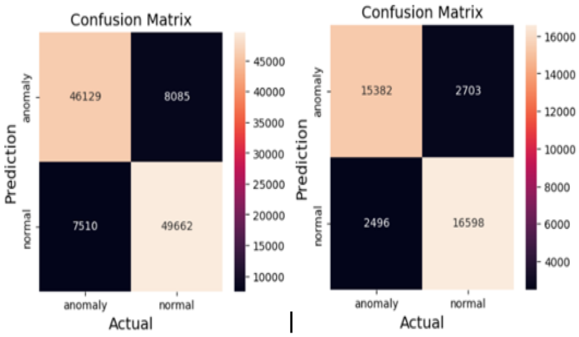
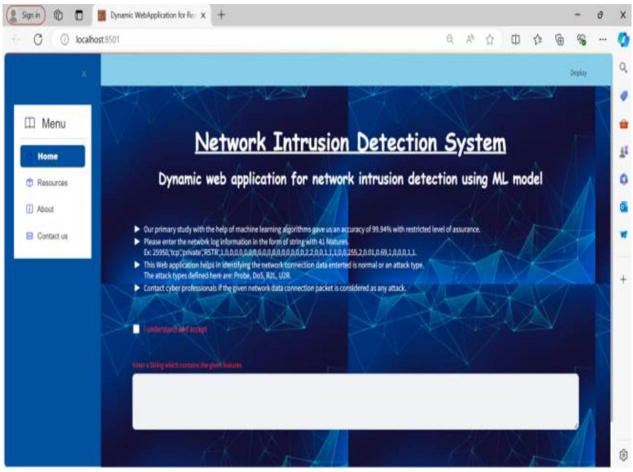


Fig.13 Confusion matrixes for train and test data of DQN model

D. Web Application Results

The web application has the following results when performed for a given data with the features. Fig.14 show the first screen of the web application that has a check box and field to enter any string value.



Network Intrusion Detection System

Dynamic web application for network intrusion detection using ML model

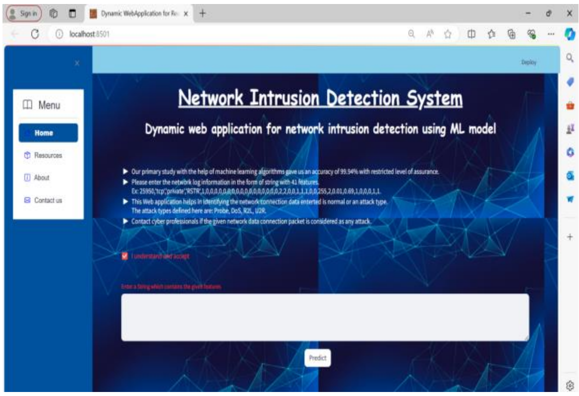
- Our primary study with the help of machine learning algorithms gave us an accuracy of 99.94% with restricted level of assurance.
- Please enter the network log information in the form of string with 41 features.
  Ex: 25950,'tcp','private','RSTR',1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,1,1,1,0,0,255,2,0.01,0.69,1,0,0,0,1,1
- This Web application helps in identifying the network connection data enterted is normal or an attack type.
  The attack types defined here are: Probe, DoS, R2L, U2R.
- Contact cyber professionals if the given network data connection packet is considered as any attack.

□ I understand and accept

Enter a String which contains the given features

Fig.14 First screen of the web application

In the web application when the check box is checked then a predict button will appear as shown in the fig.15.



Network Intrusion Detection System

Dynamic web application for network intrusion detection using ML model

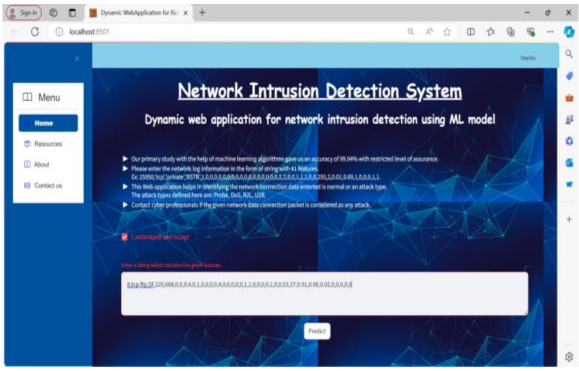
- Our primary study with the help of machine learning algorithms gave us an accuracy of 99.94% with restricted level of assurance.
- Please enter the network log information in the form of string with 41 features.
  Ex: 25950,'tcp','private','RSTR',1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,1,1,1,0,0,255,2,0.01,0.69,1,0,0,0,1,1.
- This Web application helps in identifying the network connection data enterted is normal or an attack type.
  The attack types defined here are: Probe, DoS, R2L, U2R.
- Contact cyber professionals if the given network data connection packet is considered as any attack.

☑ I understand and accept

Enter a String which contains the given features

Fig.15 First screen with the prediction button

The fig.16, 17, 18, 19 shows the accurate results according to the given input. The results of the web application contains attack type, description, causes and recommendation fields that tell about the given network connection data.



Network Intrusion Detection System

Dynamic web application for network intrusion detection using ML model

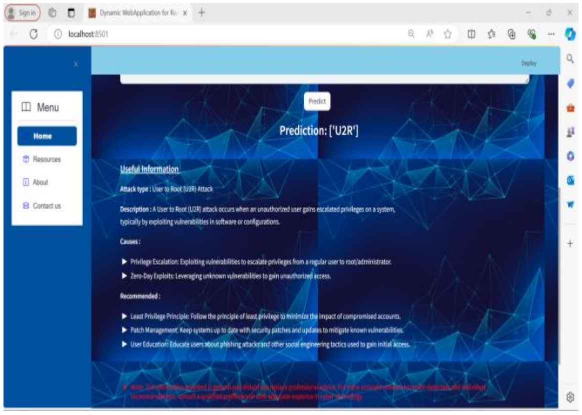
- Our primary study with the help of machine learning algorithms gave us an accuracy of 99.94% with restricted level of assurance.
- Please enter the network log information in the form of string with 41 features.
  Ex: 25950,'tcp','private','RSTR',1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,1,1,1,0,0,255,2,0.01,0.69,1,0,0,0,1,1.
- This Web application helps in identifying the network connection data enterted is normal or an attack type.
  The attack types defined here are: Probe, DoS, R2L, U2R.
- Contact cyber professionals if the given network data connection packet is considered as any attack.

☑ I understand and accept

Enter a String which contains the given features

```
0,tcp,ftp,SF,220,688,0,0,0,4,0,1,0,0,0,0,4,0,0,0,0,0,1,1,0,0,0,0,1,0,0,53,27,0.51,0.08,0.02,0,0,0,0,0
```

Fig.16 String containing the features of U2R attack type



Prediction: ['U2R']

Useful Information

Attack type : User to Root (U2R) Attack

Description : A User to Root (U2R) attack occurs when an unauthorized user gains escalated privileges on a system, typically by exploiting vulnerabilities in software or configurations.

Causes :

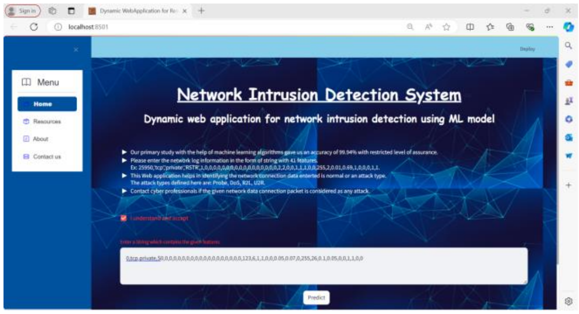
- Privilege Escalation: Exploiting vulnerabilities to escalate privileges from a regular user to root/administrator.
- Zero-Day Exploits: Leveraging unknown vulnerabilities to gain unauthorized access.

Recommended :

- Least Privilege Principle: Follow the principle of least privilege to minimize the impact of compromised accounts.
- Patch Management: Keep systems up to date with security patches and updates to mitigate known vulnerabilities.
- User Education: Educate users about phishing attacks and other social engineering tactics used to gain initial access.

Note: The information provided is general and should not replace professional advice. For more accurate [illegible] detection and individual recommendations, consult a qualified professional with adequate experience in cyber security.

Fig.17 Results of prediction of the string to U2R



Network Intrusion Detection System

Dynamic web application for network intrusion detection using ML model

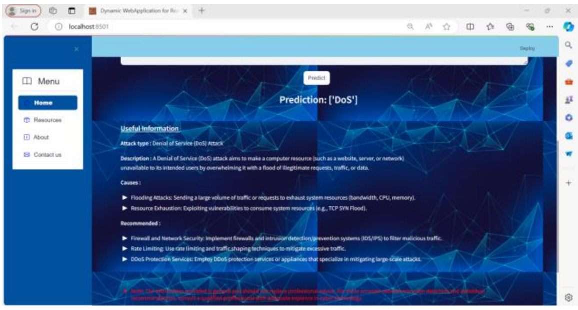
- Our primary study with the help of machine learning algorithms gave us an accuracy of 99.94% with restricted level of assurance.
- Please enter the network log information in the form of string with 41 features.
  Ex: 25950,'tcp','private','RSTR',1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,1,1,1,0,0,255,2,0.01,0.69,1,0,0,0,1,1.
- This Web application helps in identifying the network connection data enterted is normal or an attack type.
  The attack types defined here are: Probe, DoS, R2L, U2R.
- Contact cyber professionals if the given network data connection packet is considered as any attack.

☑ I understand and accept

Enter a String which contains the given features

```
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1,1,0,0,0.05,0.07,0,255,26,0.1,0.05,0,0,1,1,0,0
```

Fig.18 String containing the features of DoS attack type



Prediction: ['DoS']

Useful Information

Attack type : Denial of Service (DoS) Attack

Description : A Denial of Service (DoS) attack aims to make a computer resource (such as a website, server, or network) unavailable to its intended users by overwhelming it with a flood of illegitimate requests, traffic, or data.

Causes :

- Flooding Attacks: Sending a large volume of traffic or requests to exhaust system resources (bandwidth, CPU, memory).
- Resource Exhaustion: Exploiting vulnerabilities to consume system resources (e.g., TCP SYN Flood).

Recommended :

- Firewall and Network Security: Implement firewalls and intrusion detection/prevention systems (IDS/IPS) to filter malicious traffic.
- Rate Limiting: Use rate limiting and traffic shaping techniques to mitigate excessive traffic.
- DDoS Protection Services: Employ DDoS protection services or appliances that specialize in mitigating large-scale attacks.

Note: The information provided is general and should not replace professional advice. For more accurate [illegible] detection and individual recommendations, consult a qualified professional with adequate experience in cyber security.

Fig.19 Results of prediction of the string to DoS

E. Analysis of the Results

The ML models used in this projects shows adequate results in prediction of the classes of the given network connection data. Fig.20 shows the accuracy and roc_auc graph for different supervised learning techniques like Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR) and CatBoost algorithms. The CatBoost shows an accuracy of 99.23% for the multi-class classification. The accuracy scores are 99.99%, 99.98%, 99.98% and the roc_auc scores are 1 for RF, SVM, and LR respectively.

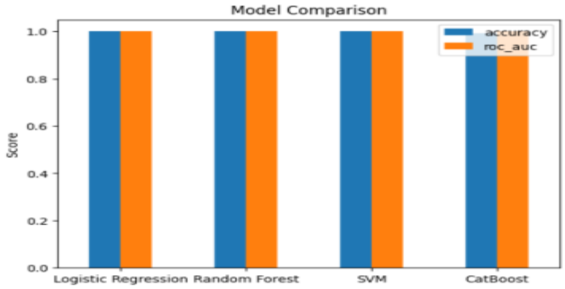


Fig.20 Accuracy and ROC_AUC values of different supervised learning models compared with CatBoost

The accuracy, precision, recall, and f1 score are found for the each ML model used in the project. Here as the unsupervised and reinforcement learning cannot predict these values, so introduced the binary classification of the attack and normal classes for the given dataset by implementing them with supervised learning algorithm. Fig.21 shows the Metrics Vs Scores graph of the used ML algorithms.

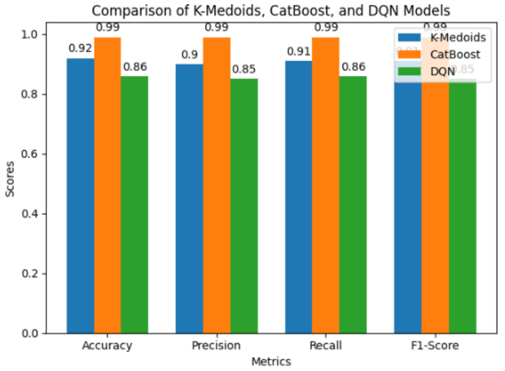


Fig.21 Metrics Vs Scores of ML model used for classifications

V. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper aims to construct a web application for predicting the network connection data packet to be a real-time threat or normal type using machine learning algorithms. As the network intrusion detection is one of the major areas of security for any system and is prominent in many organizations.

In this project, machine learning algorithms like Categorical Boosting, K-Medoids clustering and Deep Q Network are used to classify and detect the network connection to be one of the attack types that is DoS, U2R, R2L, probe or normal connection. The CatBoost algorithm gave the desired accuracy for the multi-class classification i.e. 99.24% as accuracy where as the other algorithms gave an accuracy of 92% and 86% for the binary classification which are implemented with combination of supervised algorithm. Silhouette's score for the k-medoids is plotted and the learning rate and loss of the DQN model are also plotted for better study.

Finally, a web application is constructed with the CatBoost algorithm in the back to predict a string with 41 features or less (includes basic features) to one of the attack classes or normal class and displaying the attack type, description, causes, and recommendation for solutions.

B. Future work

Future works suggested for this project include the increase the number of episodes to run for the DQN reinforcement learning, optimizing the configurations of the k-medoids and CatBoost algorithms before the model execution for the reduction of time complexity. The accuracy of the models can be improved accordingly so that it reduces false positives. The web application developed can only detects and classify into major attack categories like DoS, R2L, U2R, probe, or normal this can be overridden and classification and detection can be extend to other classes and sub classes of attacks.

REFERENCES

[1] *NSL-KDD Dataset*, NSL-KDD data set for network-based intrusion detection systems, 2009. [Online].

[2] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[3] Musa, Usman Shuaibu, et al. "Intrusion detection system using machine learning techniques: A review." *2020 international conference on smart electronics and communication (ICOSEC)*. IEEE, 2020.

[4] Tait, Kathryn-Ann, et al. "Intrusion detection using machine learning techniques: an experimental comparison." *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*. IEEE, 2021.

[5] Gokul, A., et al. "Intrusion Detection in Computer Networks Using Machine Learning Algorithms." *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, 2021.

[6] X. Zhang, J. Ran and J. Mi, "An Intrusion Detection System Based on Convolutional Neural Network for Imbalanced Network Traffic," *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, Dalian, China, 2019, pp. 456-460, doi: 10.1109/ICCSNT47585.2019.8962490.

[7] Jain, Vinay Kumar. “Network Intrusion Detection Using CatBoost Algorithm.” (2019).

[8] Hsu, Ying-Feng, and Morito Matsuoka. "A deep reinforcement learning approach for anomaly network intrusion detection system." *2020 IEEE 9th international conference on cloud networking (CloudNet)*. IEEE, 2020.

[9] Rajat Bajwa, Rasneet Kaur (2019). “Network Intrusion Detection System using Improved KMediod Clustering Approach” *IJRECE VOL. 7 ISSUE 3 JULY.-SEPT 2019*, ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE).

[10] Bahjat, Hala, et al. "Anomaly Based Intrusion Detection System Using Hierarchical Classification and Clustering Techniques." *2020 13th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE, 2020.

[11] "Feature Selection Techniques in Machine Learning." *Analytics Vidhya*, John Doe, 15 Oct. 2020, https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/.

[12] Vicky60629. *Network Intrusion Detection System*. GitHub, https://github.com/vicky60629/Network-Intrusion-Detection-System.

[13] Sharma, Anshul. "A Deeper Dive into the NSL-KDD Data Set." *Medium*, 10 Dec. 2019, https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657.

[14] Neel. "Datasets for Threat Detection Research." *Medium*, 23 Aug. 2021, https://0xneel.medium.com/datasets-for-threat-detection-research-39ee1c9b722c.

[15] OpenAI. *ChatGPT*. July version, 2024. https://chat.openai.com/

[16] Prokhorenkova, Liudmila, et al. "CatBoost: unbiased boosting with categorical features." *Advances in neural information processing systems* 31 (2018).

[17] Wang, Tiexing, et al. "K-medoids clustering of data sequences with composite distributions." *IEEE Transactions on Signal Processing* 67.8 (2019): 2093-2106.

[18] Arora, Preeti, and Shipra Varshney. "Analysis of k-means and k-medoids algorithm for big data." *Procedia Computer Science* 78 (2016): 507-512.

[19] Sewak, Mohit, and Mohit Sewak. "Deep q network (dqn), double dqn, and dueling dqn: A step towards general artificial intelligence." *Deep reinforcement learning: frontiers of artificial intelligence* (2019): 95-108.

[20] TensorFlow. "Introduction to Reinforcement Learning with TensorFlow Agents." *TensorFlow*, n.d., https://www.tensorflow.org/agents/tutorials/0_intro_rl.