CLASSIFYING MALICIOUS USER REQUEST IN THE NETWORK USING DEEP LEARNING

1. Sai Ritvik Bokka, Department of CSE, Gandhi Institute of Technology and Management, Telangana, India, Sairitvik2002@gmail.com

2. Harshith Reddy Meda, Department of CSE, Gandhi Institute of Technology and Management, Telangana, India. Harshith18@yahoo.com

Abstract: This study investigates how different machine learning strategies can be utilized to sort destructive client demands in an organization. Cyber dangers are getting more brilliant, so it's critical to have a decent attack warning system . We take a gander at four distinct methodologies — Linear Regression, Naive Bayes, XGBoost, and Convolutional Neural Networks (CNN) — to perceive how well they work at finding and arranging harmful network traffic. As standard methods, Linear Regression and Naive Bayes are utilized as specific illustrations. XGBoost, then again, utilizes a gradient boosting method. CNN, then again, utilizes deep learning out how to take out helpful attributes from network information naturally. Our review sees how well, rapidly, and generally these techniques can differentiate among threatening and safe requests, demonstrating the way that valuable they could be, in actuality, network security situations.

Index Terms – *Malicious Users, Machine Learning, Deep Learning, Linear Regression, Naïve Bayes, XGboost, CNN.*

1. INTRODUCTION

In the constantly changing world of digital networks, it's critical to have the option to effectively sort and block hurtful user demands. Cyber dangers are getting more brilliant, which shows how significant solid breach detection systems are for keeping networks safe and data safe. This study takes a gander at four different AI strategies for distinguishing harmful network data. These are Linear Regression, Naive Bayes, XGBoost, and Convolutional Neural Networks (CNN).

Linear Regression and Naive Bayes are normal strategies that can be utilized as starting stages for a full survey of performance. A gradient boosting method called XGBoost opens up additional choices by improving grouping. Convolutional Neural Organizations utilize the force of deep learning simultaneously, consequently taking out significant attributes from network information.

The main purpose of this study is to look at how well, quickly, and accurately these different methods can find and classify harmful user requests. This study aims to make a big difference in the ongoing effort to strengthen network defenses against the constantly changing scene of cyber dangers by showing how these ideas can be used in real-life network security situations.

The point of this study is to find out how well Linear Regression, Naive Bayes, XGBoost, and Convolutional Neural Networks can sort harmful user requests in networks. Linear Regression and Naive Bayes are used as standards and compared to CNN's deep learning and XGBoost's gradient boosting. The study looks at how accurate, efficient, and scalable the system is to find out how it can be used in real-life network security situations.

This study looks at how well four machine learning techniques—Linear Regression, Naive Bayes, XGBoost, and Convolutional Neural Networks—can sort harmful user requests in networks. This is done because cyber dangers are getting smarter all the time. With the need for a good breach detection system, the study aims to find out how accurate, efficient, and scalable they are so that they can be used to improve network security in the real world.

2. LITERATURE REVIEW

Developing a Deep Learning Model to Classify Malicious User on Social Networks:

Bad behavior has gotten worse since more people use social media for work and politics. People have been influenced to do things that may not be in their best interests by personal, business, and political propaganda on social media. Bad users have been able to use usernames that have been hacked to spread criticism with little to no consequences. The study's goal is to find any problems by looking at important parts of social media data and finding complicated links that can show bad behavior like fake or harmful social network accounts. In a social setting, the study uses Influence, Homophile, and Balance Theory to improve the accuracy of identifying dangerous users. The Jaccard coefficient is used to find out how similar two things are, and graphics and verbal cues are used in User-space to sort end users into groups. Standard criteria, such as the confusion matrix, are used to test the structure and see how well it works. The friend link recognition scheme is suggested for finding strange things in social atoms.

Malicious-Traffic Classification Using Deep Learning with Packet Bytes and Arrival Time:

As computers get better, Internet technology is also getting better very quickly. On the other hand, we haven't had any problems like viruses with these changes. For years, different ways to find malware have been studied as a way to deal with harmful codes. Traffic can be put into three main groups. One is based on ports, one on payloads, and the third on machine learning. We use CNN, a deep learning system, to try to sort bad traffic into different groups. For CNN, the amount of the packet and the time it arrived are what we use. The amount of the packet and its arrival time are taken out, and the data is then turned into a picture file. CNN then uses the changed picture to figure out what kind of attack the traffic is. There was 95% success in the suggested method, which was very good and showed that sorting was possible.

Classification of Malicious URLs Using Machine Learning:

Every day, hundreds of new websites make it tougher to differentiate safe from harmful ones. Websites regularly capture user data. Lack of security measures, such as swiftly discovering and flagging malicious URLs, might compromise users' privacy. The work aims to construct machine learning models that swiftly discover and categorize dangerous URLs

to make the internet safer. Bayesian optimization, SVMs, RFs, DTs, and KNNs are used to accurately categorize URLs in this work. Some instance selection techniques speed calculations. Random selection, DRLSH-based data reduction, and BPLSH-based border point extraction are these. The findings show that RFs have strong accuracy, memory, and F1 scores. SVMs perform well but take longer to train. The findings also reveal that the instance selection strategy greatly affects these models' performance, demonstrating its importance in machine learning for poor URL classification.

Malicious Web Request Detection Using Character-level CNN:

Web input injection attacks are common and powerful. Bad individuals may attack websites by introducing destructive code to HTTP requests. Many Web Intrusion Detection Systems (WIDS) can't detect undiscovered new attacks and have a high FPR for web parameter injection attacks. Because they can't learn from errors and don't care about character relationships. Our paper proposes a superior convolution neural network (CNN) model for discovering bad requests that can learn from failures. Before the convolution layer, we add a character-level embedding layer to assist our model understand how the query string letters relate. We also update CNN's filters to extract the query string's finer details. The test results reveal that our model has a lower FPR than SVM and RF.

Deep Learning Models for Malicious Web Content Detection: An Enterprise Study:

Web malware detection is still crucial to enterprise security. Heuristics-based recognition models that need feature engineering may not scale adequately. This project sorts 800,000 URL HTML strings into a fair list of "bad" and "good." We employ preprocessing approaches to acquire valuable information into a Bag-of-Tokens representation and develop convolutional and sequenced-based deep learning models for binary classification. These models perform well on balanced datasets, with the top model averaging 95% accuracy. We train models with various loss functions and change the bad-to-good ratio to simulate zero-day attacks. These models follow heuristics-based concepts in corporate cybersecurity. With 50% and 25% accuracy on malicious: neutral rates of 1:10 and 1:00, respectively, they barely succeed.

3. METHODOLOGY

Currently, network security mostly uses rule-based and signature-based methods, like firewalls and intrusion detection systems (IDS), to sort hostile user requests into different groups. These systems use rules and patterns that have already been set up to find known dangers. This makes them good at spotting well-known attack patterns. Still, it's hard for them to adapt to new and changing dangers. Even though machine learning models are becoming more popular, they are often used with other tools and can't handle class mismatch and dynamic attack scenarios on their own. Also, the current systems might not be flexible enough to handle large amounts of data quickly and effectively, which is very important for network security. In this case, the suggested system tries to get around these problems by using deep learning and advanced data preparation techniques to make the attack detection process more accurate and flexible.

Drawbacks:

1. The ongoing strategy depends on decides and signs that have proactively been set, which makes it less flexible when dangers change or are unknown. It struggles with finding new attack patterns or changes.

2. Older systems don't work effectively of managing class contrasts, which could cause a ton of fake positives or miss important threats.

3. The current systems probably won't have the option to deal with a lot of data quickly and efficiently, which is significant for meeting the security needs of the present organizations.

The proposed system for identifying harmful user requests on a network is a complete method that includes several important steps, such as Exploratory Data Analysis (EDA), using an algorithm, and then comparing how well each algorithm worked. During the EDA process, the data is first checked for null numbers to make sure it is correct. Data display methods are used to learn more about the features of the information. To improve the quality of the raw data, columns that aren't needed are taken out and category mapping is used. Feature selection makes the sample better for training the model. The Synthetic Minority Over-sampling Technique (SMOTE) is used to fix problems with class mismatch. Linear Regression, Naive Bayes, XGBoost, and Convolutional Neural Networks (CNN) are the four methods that are looked at in the algorithm application process. Each method is tested to see how well it can correctly classify harmful user requests. The last part of the comparison tests how well these algorithms find and classify harmful requests. This gives us useful information for making a strong attack detection system for network security.

Benefits:

1. The suggested system uses Exploratory Data Analysis (EDA) and advanced data preparation methods to make the data more reliable and improve the quality of the data that is input.
2. The suggested system is better able to react to changing danger scenarios because it uses deep learning methods to learn and spot new attack patterns more quickly.
3. The Synthetic Minority Over-sampling Technique (SMOTE) is used to fix class mismatches, which lowers the chance of false positives and makes it easier to find threats.
4. The suggested system carefully evaluates various algorithms, letting the best and most accurate way to identify fraudulent requests be chosen, thereby improving the intrusion detection system as a whole.

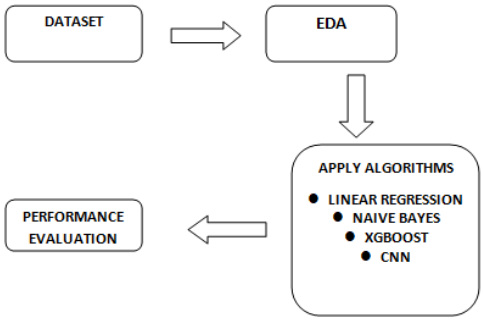

Fig 1 System Architecture

MODULES:

To carry out the above job, we have created the following modules:

1. Importing Libraries:

In this step, you'll bring in the libraries you need to work with data, display it, and train machines. Pandas, NumPy, Matplotlib/Seaborn, and scikit-learn are all common tools for machine learning methods and graphics.

2. Importing Dataset:

Add the dataset to your system. This step is very important if you want to understand the info you'll be working with.

3. EDA (Exploratory Data Analysis):

- ✓ *Checking for null values:* Find and deal with missing data to keep the dataset's integrity.
- ✓ *Data Visualization:* Use tools like histograms, scatter plots, and heatmaps to make pictures that show how the data is distributed and how it is related to other data.
- ✓ *Removing unwanted columns:* Get rid of features that don't add anything to the study or could cause noise.
- ✓ *Category mapping:* If your information has category factors, you need to turn them into number values for machine learning methods.
- ✓ *Feature Selection:* Pick out useful traits to make the model work better.
- ✓ *SMOTE (Synthetic Minority Over-sampling Technique):* To even out the information when you have classes that aren't fair, use methods like SMOTE.

4. Applying Algorithms:

- ✓ *Linear Regression:* Use a linear regression model to solve your problem and rate it.
- ✓ *Naive Bayes:* Use Naive Bayes and rate how well it works.
- ✓ *XGBoost:* Use the gradient boosting method XGBoost and look at how well it works.
- ✓ *CNNs (Convolutional Neural Networks):* Use a CNN to do things like sorting images into groups or analyzing sequences.

5. Comparison:

Check how well each method works by looking at its accuracy, precision, memory, and F1 score.

4. IMPLEMENTATION

The following methods were used in this project:

Linear Regression: This is a way to figure out what will happen in the future by finding a straight line between two variables, one that is independent and the other that is dependent. It is a way to use statistics to make predictions that is used in data science and machine learning.

Naïve Bayes: The Naïve Bayes classifier is a guided machine learning method used for jobs like text classification that involve putting things into groups. To add to that, it is a generative learning algorithm, which means it tries to model how data of a certain class or group are spread out.

XGBoost is a boosting algorithm that uses bagging to train several decision trees and then blends the outcomes. It helps XGBoost learn faster than other algorithms and is especially useful when there are a lot of factors to consider.

CNN: A CNN is a type of network design for deep learning algorithms. It is used to recognize images and do other jobs that require processing pixel data. There are different kinds of neural networks used in deep learning, but CNNs are the best for finding and recognizing things.

5. EXPERIMENTAL RESULTS

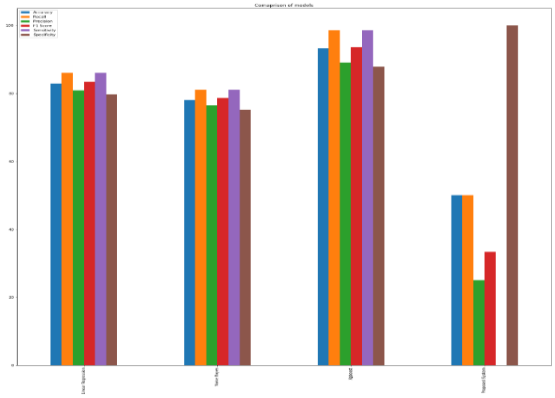

Fig 2 Comparison graph of all algorithms

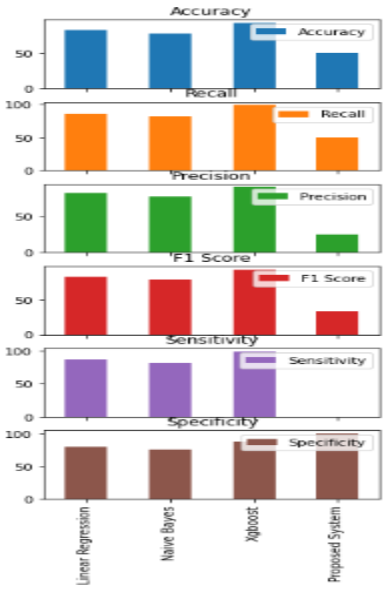

Fig 3 Performance evaluation graphs of all algorithms

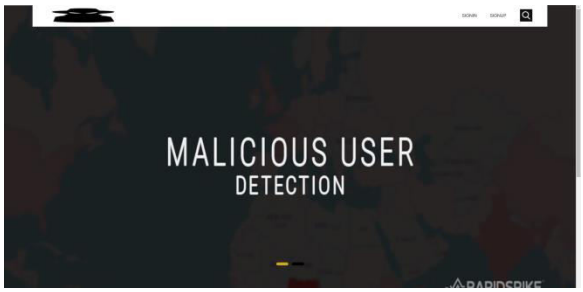

Fig 4 Home page

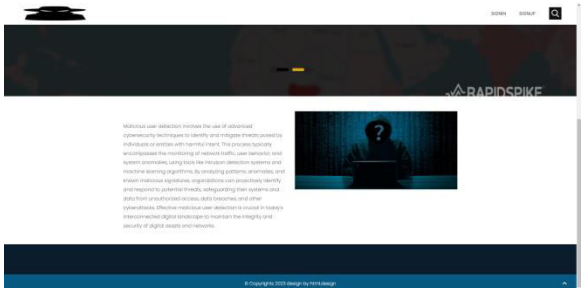

Fig 5 About page

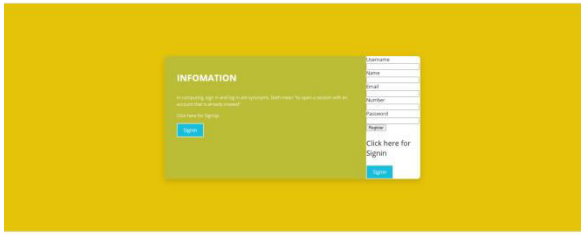

Fig 6 Registration page

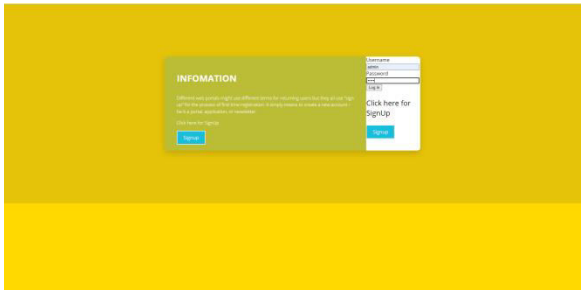

Fig 7 Login page

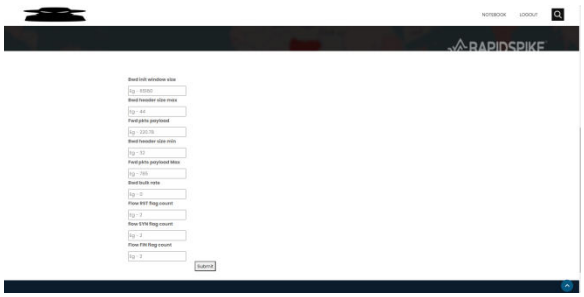

Fig 8 Main page

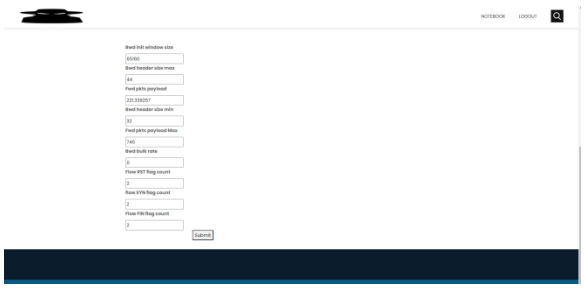

Fig 9 User upload input values to predict output

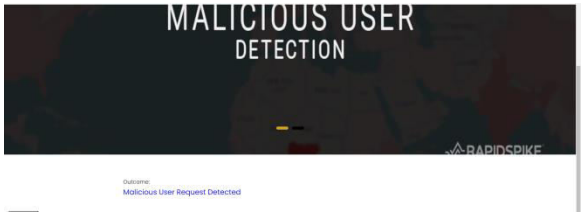

Fig 10 final outcome is malicious user request detected

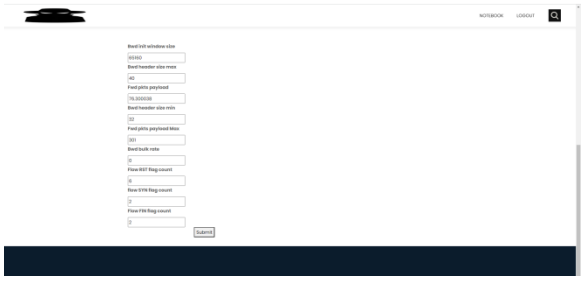

Fig 11 User upload another input values for outcome

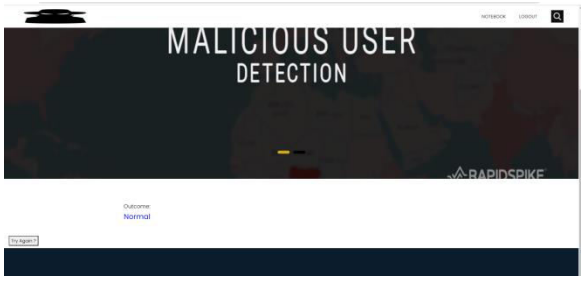

Fig 12 Predict result is normal

6. CONCLUSION

In conclusion, the suggested method for sorting harmful user requests in a network is a great step forward in the area of network security. It fixes some major problems with the current rule-based and signature-based approaches, mainly the fact that they aren't very flexible and have trouble with class mismatches and growth. Exploratory Data Analysis (EDA), improved data preparation, and the use of deep learning methods make the suggested system a better and more flexible way to find intrusions. The system uses algorithms like Linear Regression, Naive Bayes, XGBoost, and Convolutional Neural Networks (CNN) and regularly checks how well they work. This lets it choose the most accurate and time-saving way to spot false requests. This all-around method makes networks safer by making them more flexible, accurate, and able to handle big amounts of data. Since network dangers are always changing, this system is ready to be a key part of keeping digital networks safe from new security problems.

7. FUTURE SCOPE

In the future, this system might get better by adding real-time danger information, automatic model updating, and methods for finding things that don't seem right. There are also exciting options like using edge computing to speed up reaction times and

looking into quantum computing to make danger analysis even more accurate.

REFERENCES

[1] Samant Verma; Shailja Shukla “Developing a Deep Learning Model to Classify Malicious User on Social Networks” in 2023 International Conference on Communication, Circuits, and Systems (IC3S) IEEE.

[2] Ingyom Kim and Tai-Myoung Chung “Malicious-Traffic Classification Using Deep Learning with Packet Bytes and Arrival Time” in Biomedical Signal Processing, Statistical Learning, 2020.

[3] Shayan Abad, Hassan Gholamy and Mohammad Aslani “Classification of Malicious URLs Using Machine Learning” in Journals , Sensors, Volume 23 , Issue 18 , 10.3390/s23187760.

[4] Wei Rong, Bowen Zhang, Xixiang Lv “Malicious Web Request Detection Using Character-level CNN” in Semantic Scholar, DOI:10.1007/978-3-030-30619-9_2, Corpus ID: 53747021

[5] Matthew Chun Kit Wong “Deep Learning Models for Malicious Web Content Detection: An Enterprise Study” in https://tspace.library.utoronto.ca/bitstream/1807/98465/1/Wong_Matthew_%20_201911_MAS_thesis.pdf

[6] J. Zhang, B. Dong and P. S. Yu, "Deep Diffusive Neural Network based Fake News Detection from Heterogeneous Social Networks", 2019 IEEE International Conference on Big Data (Big Data), pp. 1259-1266, 2019.

[7] Z. Shahbazi and Y.-C. Byun, "Fake Media Detection Based on Natural Language Processing and Blockchain Approaches", IEEE Access, vol. 9, pp. 128442-128453, 2021.

[8] Z. Yang, J. Xue, X. Yang, X. Wang and Y. Dai, "VoteTrust: Leveraging Friend Invitation Graph to Defend against Social Network Sybils", IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 4, pp. 488-501, July-Aug. 2016.

[9] R. Barbado, O. Araque and C. A. Iglesias, "A framework for malicious review detection in online consumer electronics retailers", Information Processing and Management, vol. 56, no. 4, pp. 1234-1244, 2019.

[10] Y. Liu, B. Pang and X. Wang, "Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph", *Neurocomputing*, vol. 366, pp. 276-283, 2019.

[11] J. K. Rout, A. K. Dash and N. K. Ray, "A framework for malicious review detection: Issues and challenges", 2018 International Conference on Information Technology (ICIT), pp. 7-10, 2018.

[12] Fuzhi Zhang, Shuai Yuan, Peng Zhang, Jinbo Chao and Hongtao Yu, "Detecting review spammer groups based on generative adversarial networks", *Information Sciences*, vol. 606, pp. 819-836, 2022, ISSN 0020-0255.

[13] Jun Yin, Qian Li, Shaowu Liu, Zhiang Wu and Guandong Xu, "Leveraging multi-level dependency of relational sequences for social spammer detection", *Neurocomputing*, vol. 428, pp. 130-141, 2021, ISSN 0925-2312.

[14] Nour El-Mawass, Paul Honeine and Laurent Vercouter, "Enhanced social spammers detection on Twitter using Markov Random Fields", *Information Processing & Management*, vol. 57, no. 6, pp. 102317, 2020, ISSN 0306-4573.

[15] Sarah Riddick and Rich Shivener, "Affective Spamming on Twitch: Rhetorics of an Emote-Only Audience in a Presidential Inauguration Livestream", *Computers and Composition*, vol. 64, pp. 102711, 2022, ISSN 8755-4615.

[16] Vishnu Dutt Sharma, Santosh Kumar Yadav, Sumit Kumar Yadav, Kamakhya Narain Singh and Suraj Sharma, "An effective approach to protect social media account from spam mail – A machine learning approach", *Materials Today: Proceedings*, 2021, ISSN 2214-7853.

[17] Raja kumari Mukiri and B. Vijaya Babu, "Prediction of rumour source identification through spam detection on social Networks- A survey", *Materials Today: Proceedings*, 2021, ISSN 2214-7853.

[18] Zhiwei Guo, Lianggui Tang, Tan Guo, Keping Yu, Mamoun Alazab and Andrii Shalaginov, "Deep Graph neural network-based spammer detection under the perspective of heterogeneous cyberspace", *Future Generation Computer Systems*, vol. 117, pp. 205-218, 2021, ISSN 0167-739X.

[19] Aaisha Makkar, "SecureEngine: Spammer classification in cyber defence for leveraging green computing in Sustainable city", *Sustainable Cities and Society*, vol. 79, pp. 103658, 2022, ISSN 2210-6707.

[20] Hyungho Byun, Sihyun Jeong and Chong-kwon Kim, "SC-Com: Spotting Collusive Community in Opinion Spam Detection", *Information Processing & Management*, vol. 58, no. 4, pp. 102593, 2021, ISSN 0306-4573.