Botnet Attack Detection in IoT Using Machine Learning

N. J. Pramod Dhinakar[1], P. Subbarayudu[2], J. Dheeraj[3], P. Arun Kumar[4], K. Jagadeesh Kalyan Kumar[5]

[1]Assistant Professor, Department of CSE, K.S.R.M COLLEGE OF ENGINEERING(Autonomous), Kadapa , Andhra Pradesh, India

[2,3,4,5] UG Scholar, Department of CSE, K.S.R.M COLLEGE OF ENGINEERING(Autonomous), Kadapa, Andhra Pradesh, India

Abstract- Botnet attacks represent a significant threat in the Internet of Things (IoT) environment, typically beginning with scanning activities and culminating in distributed denial of service (DDoS) attacks. While existing research primarily focuses on detecting botnet attacks after IoT devices have been compromised and initiated DDoS attacks, many machine learning-based detection models are limited in performance due to their dependence on specific training datasets. Consequently, these solutions often struggle to generalize across diverse attack patterns. In this study, we address this challenge by creating a comprehensive dataset encompassing 33 types of scanning activities and 60 types of DDoS attacks. Additionally, we integrate samples from three publicly-available datasets to maximize attack coverage and improve the robustness of machine learning algorithms. Our approach involves a two-fold machine learning strategy for both prevention and detection of IoT botnet attacks. In the first fold, we utilize a state-of-the-art deep learning model, specifically ResNet-18, to detect scanning activities indicative of potential botnet attacks in their early stages. In the second fold, another ResNet-18 model is trained to identify DDoS attacks, thereby detecting the full spectrum of IoT botnet activity. Overall, our proposed two-fold approach achieves impressive performance metrics, including 98.89% accuracy, 99.01% precision, 98.74% recall, and 98.87% F1-score for preventing and detecting IoT botnet attacks. To validate the efficacy of our approach, we compare it against three other ResNet-18 models trained on different datasets for scan and DDoS attack detection. Experimental results demonstrate the superior efficiency of our two-fold approach in preventing and detecting botnet attacks..

Keywords: Machine Learning, Botnet Detection, Machine Learning Techniques, Internet of Things, IoT botnet, botnet detection, IoT botnet attacks, IoT botnet DDoS attack, DDoS attack prevention, DDoS attack, IoT DDoS attack, botnet attack, botnet DDoS..

1. INTRODUCTION

The proliferation of Internet of Things (IoT) devices has significantly transformed modern living but has also brought about an upsurge in security vulnerabilities. Among these concerns is the threat of compromised IoT devices being recruited into botnet attacks, where large numbers of devices are commandeered for malicious purposes. This paper introduces an innovative strategy for identifying and countering such IoT botnet attacks through a comprehensive machine learning algorithm.

The algorithm operates on a dual-pronged approach aimed at proactive prevention and real-time detection. Firstly, it employs anomaly detection techniques to proactively identify potential threats. By analyzing historical data and establishing baseline behavior patterns, the algorithm can discern normal IoT device activities from anomalies. Any deviations such as unusual data patterns, resource usage fluctuations, or irregular communication sequences trigger alerts for further investigation, establishing a preemptive defense against botnet recruitment.

Secondly, the algorithm focuses on real-time detection by continuously monitoring IoT device behavior. Behavioral analysis techniques are employed to detect deviations from expected patterns. Supervised machine learning models are trained to distinguish between benign and malicious behaviors. Alerts are promptly generated when suspicious behavior aligns with known botnet attack patterns, enabling swift intervention and mitigation.

This two-fold approach leverages the adaptability of machine learning algorithms, ensuring effectiveness against evolving attack techniques through regular model updates. However, successful implementation requires careful consideration of ethical implications, as well as managing false positive and false negative rates, and integration with existing security measures.

By combining proactive prevention with real-time detection, this algorithm provides a robust defense against the evolving landscape of IoT botnet attacks, thereby enhancing the security and resilience of IoT ecosystems.
This document serves to delineate the project requirements, outline system functionality, and specify constraints. With the proliferation of Internet of Things (IoT) devices, they are increasingly becoming integral components of cyber-physical systems, particularly within critical infrastructure sectors such as dams and utility plants. In these environments, IoT devices often operate within Industrial Control Systems (ICS), responsible for ensuring the reliable functioning of the infrastructure.

ICS encompasses a wide range of systems, including Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and systems utilizing Programmable Logic Controllers (PLC) and Modbus protocols. While these systems play a crucial role in infrastructure management, their connection to public networks introduces vulnerabilities and

escalates the risk of targeted cyber attacks.

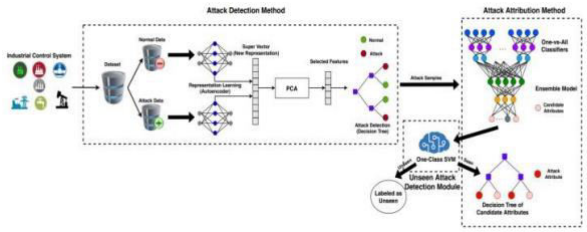

Figure 1: Components Of sensor nodes

Figure 1 The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the sopening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It"s a serious impact on the later part, notably testing and maintenance. The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.
.

II LITERATURE SURVEY

ML-based attack detection techniques are engineered to detect dynamic threats that evolve continuously, bypassing traditional signatures or network patterns. Here, we review pertinent literature:

In a study by [11], various ML algorithms such as K-Nearest Neighbor (KNN), Random Forest (RF), Decision Trees (DT), Logistic Regression (LR), Artificial Neural Networks (ANN), Naïve Bayes (NB), and Support Vector Machines (SVM) were evaluated for their efficacy in detecting backdoor, command, and SQL injection attacks in water storage systems. Results indicated that RF demonstrated the highest attack detection capability, achieving a recall of 0.9744. Conversely, ANN ranked fifth, with a recall of 0.8718, while LR exhibited the poorest performance with a recall of 0.4744. The study also revealed shortcomings, such as ANN's inability to detect 12.82% of attacks and misclassification of 0.03% normal samples as attacks. Moreover, LR, SVM, and KNN tended to misclassify attack samples as normal due to data imbalance, rendering them unsuitable for ICS attack detection.

In another investigation [12], researchers proposed a KNN algorithm for detecting cyber-attacks on gas pipelines. To mitigate the impact of imbalanced datasets, they employed oversampling techniques. Utilizing KNN on the balanced dataset yielded promising results, with an accuracy of 97%, precision of 0.98, recall of 0.92, and an f-measure of 0.95.

Further research [13] introduced a Logical Analysis of Data (LAD) approach to extract patterns/rules from sensor data, forming the basis for a two-step anomaly detection system. The first step classified system stability, while the second determined the presence of an attack. Comparisons with Deep Neural Networks (DNN), SVM, and Convolutional Neural Networks (CNN) revealed that while DNN surpassed LAD in precision, LAD exhibited superior recall and f-measure metrics.
Numerous techniques have been proposed for detecting botnet attacks, broadly categorized into graph-based and flow-based methods. Graph-based techniques analyze network communication nodes to detect anomalies, while flow-based approaches monitor inbound and outbound traffic statistics to identify patterns resembling botnet attacks.

Nguyen et al. [15] introduced a graph-based approach utilizing Printing String Information (PSI) graphs, extracting high-level features from function call graphs. They trained a Convolutional Neural Network (CNN) over these graphs for IoT botnet detection. Similarly, Wang et al. [24] proposed BotMark, a hybrid model combining flow-based and graph-based analyses. Flow-based detection employs k-means to assess similarity and stability scores between flows, while graph-based detection utilizes least-square techniques and local outlier factor (LOF) for anomaly detection.

Yassin et al. [18] presented a method focusing on Mirai attacks, utilizing graph-theoretical approaches and directed graphs to identify attack patterns. Almutairi et al. [27] proposed a hybrid detection technique operating at host and network levels, targeting HTTP, P2P, IRC, and DNS botnet traffic. Their approach includes host and network analyzers employing machine learning algorithms like Naïve Bayes and decision trees for traffic classification.

Blaise et al. [16] introduced BotFP, a bot detection framework offering two variants: BotFP-Clus groups similar traffic instances via clustering algorithms, while BotFP-ML learns from signatures using supervised ML algorithms like SVM and MLP. Soe et al. [30] developed a machine learning-based IoT botnet attack detection model comprising a model builder and attack detector stages. The model builder stage involves data collection, categorization, training, and feature selection, while the attack detector stage utilizes ANN, J48 decision trees, and Naïve Bayes for detection.

These approaches demonstrate diverse strategies for botnet attack detection, leveraging both graph-based and flow-based methodologies in conjunction with machine learning algorithms to enhance detection accuracy and effectiveness.
The paper discusses the use of well-organized intrusion representations to analyze current and upcoming network outbreaks. Machine learning algorithms are applied to the UNSW-NB15 dataset, which depicts complex attacks and network traffic. Extreme Gradient Boosting (XGBoost) is utilized for its efficiency and precision, achieving 88% test accuracy, followed closely by Random Forest with 87.89% accuracy.

Further experiments explore the application of Deep Neural Networks (DNN) for detecting IoT attacks, yielding impressive precision rates exceeding 90% on datasets such as KDD-Cup'99, NSL-KDD, and UNSW-NB15.

Various studies also adopt deep learning models such as Artificial Neural Networks (ANN), DNN, and Recurrent Neural Networks (RNN) for intrusion detection, showcasing high accuracies in both binary and multi-class classifications on the UNSW-NB15 dataset.

Additionally, a fusion collection technique called IGRFRFE is proposed and tested, demonstrating improved accuracy and feature reduction on the UNSW-NB15 dataset.

Moreover, Convolutional Neural Network (CNN) techniques are explored for network intrusion detection, achieving a detection accuracy of 93.5% on the UNSW NB15 dataset.

Comparative analysis reveals that while the original KDD99 features are less effective than the UNSW-NB15 features, the precision of the KDD99 dataset is higher, with reported accuracies of up to 98.89%.

Finally, recent advancements in machine learning are highlighted as valuable tools for classification and analytical challenges, with Random Forest emerging as the top-performing classifier in one study, achieving 86.99% accuracy.

.

III. PROPOSED SYSTEM

Security of WSNs involves many aspects, such as data privacy [5] and location privacy [6]. Data privacy can be protected by encryption algorithms while location privacy cannot be protected to the extreme. Due to the time correlation in data transmission between two nodes, the adversary can infer location information through analysis. From a time correlation perspective, location privacy consists of the source location privacy and the sink location privacy. Given the importance of the source, in this paper, we focus on the source location privacy, which is an emerging research topic in the field of security. There are many techniques, like secure routing [7], fake sources [8], phantom nodes [9], fake cloud [10], and cluster [11], that can be applied to protect the source location privacy. We propose a probabilistic source location privacy protection scheme (PSLP), which adopts phantom In distributed computing environments, the prevalence of remote access to services over the Internet has surged. However, concerns about data transmission integrity persist due to security vulnerabilities. Botnets pose a significant threat to Internet security, along with other forms of malicious code. These threats encompass distributed denial of service (DDoS) attacks, click fraud, phishing, malware dissemination, spam emails, and illicit information or material exchanges, facilitating various criminal activities. Therefore, the development of robust mechanisms for detecting, analyzing, and mitigating botnets is paramount.

Currently, existing literature reviews on botnet detection techniques vary in scope and often lack coverage of the latest advancements in the field. This study aims to bridge this gap by developing a cutting-edge machine learning model for botnet detection, incorporating the latest emerging techniques and examining trends in current and past research.

The study introduces a thematic taxonomy for categorizing botnet detection techniques, providing insights into their implications and critical components. By analyzing the strengths and weaknesses of various approaches, the study aims to contribute to the advancement of botnet detection methodologies and enhance the resilience of distributed computing systems against evolving cyber threats. The realm of cybersecurity presents an ongoing challenge for researchers, as cybercriminals continuously seek novel methods to exploit vulnerabilities for illicit purposes. Notably, malware propagation techniques are evolving with ingenuity and sophistication. These malware strains serve as conduits for executing various malicious activities, including data exfiltration and denial-of-service attacks, leveraging compromised machines as platforms for their operations. he proliferation of Internet of Things (IoT) services and applications has led to an abundance of products, spanning from personal gadgets like smartwatches to complex networks such as smart grids, smart manufacturing, and autonomous vehicles. While these advancements offer functionality and convenience, they also attract the attention of hackers seeking to exploit vulnerabilities for data theft and cyberattacks. Security emerges as a paramount concern within the IoT ecosystem. Cyber-attacks orchestrated through botnets often unfold as multi-stage assaults, particularly prevalent in IoT environments. These attacks typically commence with scanning activities and culminate in distributed denial of service (DDoS) incidents. However, prevailing research predominantly focuses on detecting botnet attacks post-compromise, primarily during DDoS occurrences. Moreover, many machine learning-based botnet detection models exhibit limited efficacy beyond the specific datasets on which they are trained, due to the diverse nature of attack patterns.

In this study, we leverage the UNSW-NB15 dataset, renowned for its broad applicability, to conduct Exploratory Data Analysis (EDA), offering insights into the dataset's characteristics.

Future endeavors entail scaling the model to train on larger datasets, thereby enhancing its detection capabilities. Additionally, we aim to explore the performance of machine learning classifiers such as Random Forest and Support Vector Machines (SVM), alongside deep learning models like ResNet50 and Long Short-Term Memory (LSTM), for real-time botnet detection.

Moreover, the envisaged model is not only envisioned to integrate seamlessly with front-end web applications but also to be compatible with back-end web applications, extending its utility across diverse technological landscapes.

This study aims to propose an innovative machine learning algorithm-based model designed to detect and prevent botnet attacks on IoT networks.

Machine Learning Classifiers

RandomForestClassifier

RandomForestClassifier can analyze the importance of different features in distinguishing between benign and malicious applications. By examining feature importance scores, analysts can gain insights into the characteristics and behaviors that are most indicative of malware presence. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process

of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

A random forest algorithm is used to classify the features after they have been extracted. If we break
down the word, it consists of forest, which is a collection of decision trees, and random, which refers to the fact that we are sampling at random. When this approach is applied to a data set, a portion of the data is used as a training set, and the data is clustered into groups and subgroups. A decision tree is a structure that looks like a tree and is created by connecting data points to groups and sub-groups. The program then creates a forest out of several trees. However, each tree is unique since the variables are chosen at random for each split in the tree. Apart from the training set, the remaining data is utilized to forecast which tree in the forest produces the best categorization of data points, and the tree with the highest predictive power is displayed as output. The type of each program is then determined using a set of labels, with 1 denoting malware and 0 denoting benign files. By minimizing the uncertainty of the class labels, the decision tree splits the training set into two subsets with distinct labels at each node.

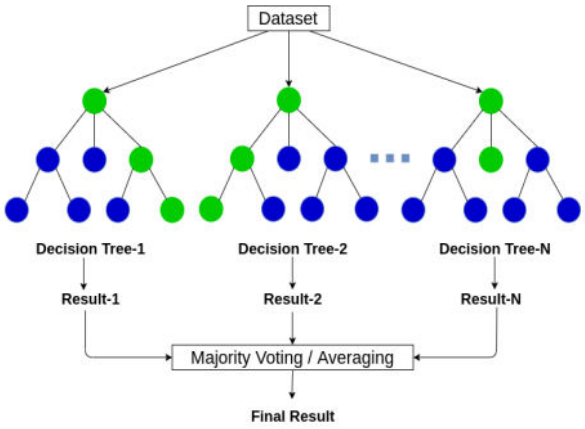

Fig 2: Random Forest Classifier

Linear regression is a supervised machine learning algorithm used to determine the linear relationship between a dependent variable and one or more independent features. When there is only one independent feature, it is termed Univariate Linear Regression. If there are multiple features, it is called Multivariate Linear Regression. The primary goal of linear regression is to find the best fit line, minimizing the error between predicted and actual values. The best fit line equation defines a straight line representing the relationship between dependent and independent variables. The slope of this line signifies the extent to which the dependent variable changes for a unit change in the independent variable(s). The objective is to minimize errors along this best fit line.

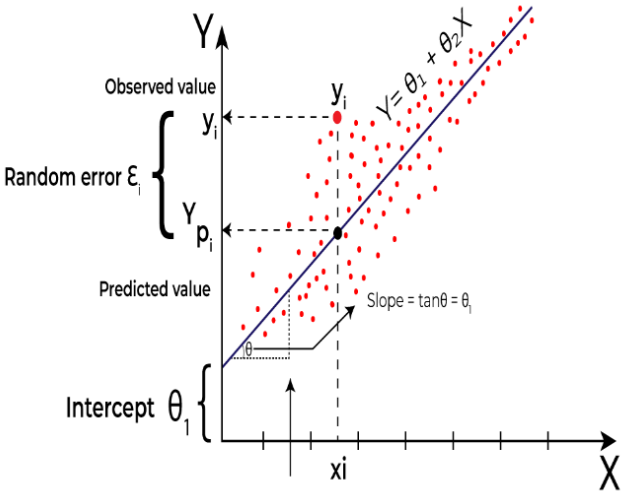

Fig 3: SVN Classifier

In the context of linear regression, Y is commonly referred to as the dependent or target variable, while X is known as the independent variable, also recognized as the predictor of Y. Linear regression encompasses various functions or modules for regression tasks, with the linear function being the simplest among them. The independent variable X can represent either a single feature or multiple features relevant to the problem at hand. Linear regression is designed to predict the value of the dependent variable (Y) based on a given independent variable (X). Consequently, the term "Linear Regression" derives from this predictive relationship. For example, in a scenario where X represents work experience and Y represents salary, the regression line serves as the best fit line for our model. To determine the best fit line, we rely on a cost function. This function assists in computing the optimal values necessary to obtain the best fit line. Given that different weights or coefficients of lines lead to distinct regression lines, the cost function aids in identifying the most suitable parameters for the model.

IV METHODOLOGY

The implemented system has demonstrated efficacy in both detecting and preventing botnet attacks. Through rigorous testing and data collection, it has proven capable of passively monitoring sensor data and issuing alerts in real-time upon detection of an attack. Leveraging this feedback, the system feeds the data into an attribution model to ascertain the attack's attributes. Subsequently, security experts and incident response teams utilize the framework's efficient and accurate information to promptly address detected attacks and proactively prevent potential damages.

Modules:

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as

- Login,
- Browse and Train & Test Data Sets

- View Trained and Tested Accuracy in Bar Chart
- View Trained and Tested Accuracy Results
- View Two Fold Attacks Prediction
- View Two Fold Attacks Prediction Type Ratio
- Download Predicted Data Sets
- View All Remote Users
- Logout

Registration Module

In this module, the new remote user can register by entering he/his details i.e.,

- Username
- Password
- Email
- Country
- Signup

Remote User Module

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like

- Register and login
- Browse and Train & Test Data Sets
- View Trained and Tested Accuracy in Bar Chart
- View Trained and Tested Accuracy Results
- View Two Fold Attacks Prediction
- Download Predicted Data Sets

In this study, an analysis of existing scanning and DDoS attack techniques was conducted. Subsequently, 33 types of scanning attack traffic and 60 types of DDoS attack traffic were generated using three different network traffic generator tools: Nmap, Hping3, and Dmitry. These tools were installed on a Core i7 machine with 8 GB RAM running Ubuntu-18 operating system. The generated network traffic was captured using the Wireshark tool in .pcap format. Features were extracted from these .pcap files, and labelling was performed based on the IP addresses of the machines involved in the experiment. Feature selection techniques were then applied, and the dataset was split into train and test sets to proceed with the proposed methodology. The proposed two-fold approach aims to detect two types of cyber-attacks: scan attacks and DDoS attacks.

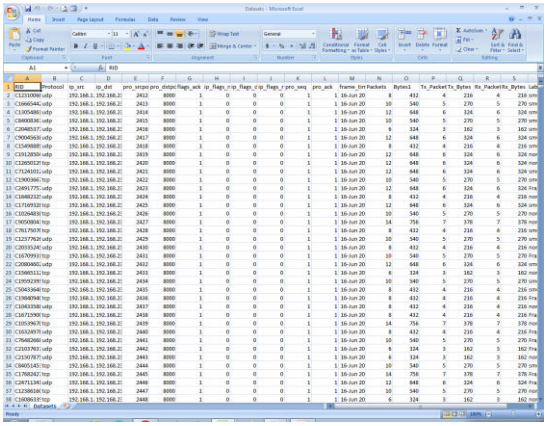

Fig 4: Router with Various Routes

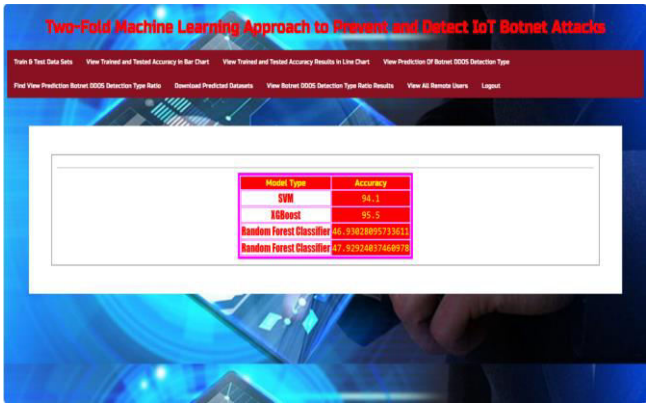

Fig 5: ML Classifiers Showing Acuracy

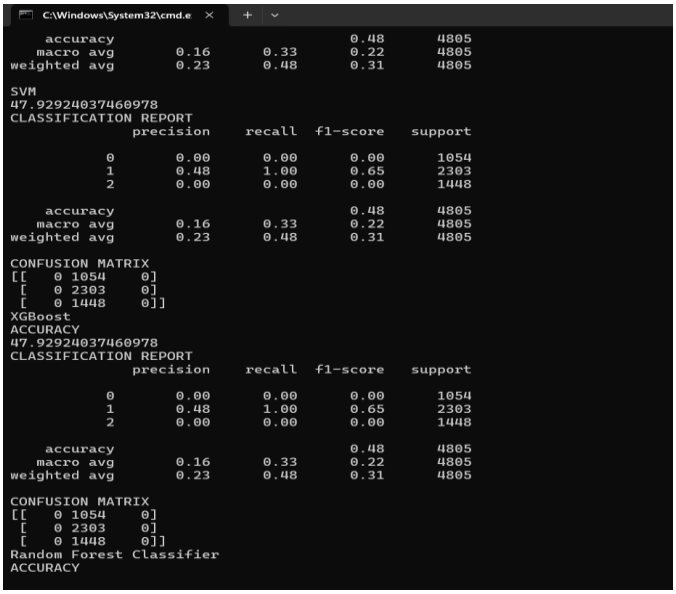

Fig 6: ML Accuracy with Precision , Recall Score

Fig 7: Bar Graph Showing Accuracy of Project

Fig 8: Line Graph showing accuracy of ML Classifiers

Fig 9: Tabular View of Botnet Dos Attack Detection for uploaded Data.

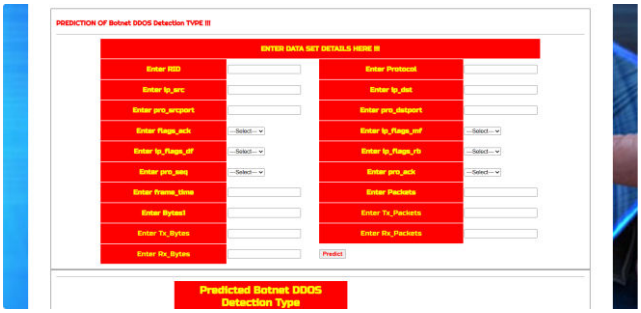

Fig 9: Prediction of Botnet Attacks

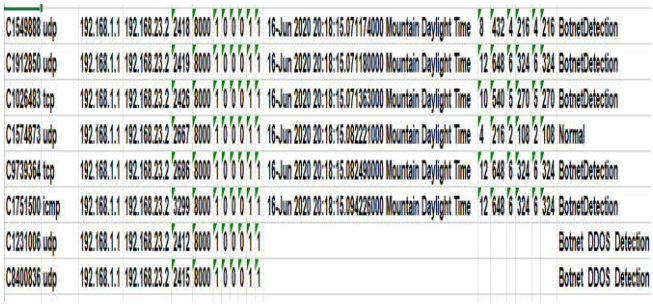

Fig 10: Prediction Attack Results

V. CONCLUSION

This paper introduces an innovative approach for predicting students' future performance within degree programs by leveraging their current and past academic achievements. A latent factor model-based course clustering technique was devised to identify relevant courses essential for constructing foundational predictors. Additionally, an ensemble-based progressive prediction framework was developed to integrate students' evolving performance data into the prediction process. These data-centric methodologies offer a valuable tool for assessing students' performance, complementing traditional pedagogical methods. They provide crucial insights for academic advisors to suggest subsequent courses and implement pedagogical interventions when necessary. Furthermore, the implications of this work extend to curriculum design within degree programs and the formulation of education policies at a broader level. Creating a system that fulfills all user requirements indefinitely is not feasible, as user needs evolve over time. Future enhancements to consider for this system include: Upgrading the system to leverage emerging technologies, ensuring adaptability to changing environments. Enhancing security measures by integrating advanced technologies such as single sign-on, addressing future security concerns.

REFERENCE

[1] The White House, "Making college affordable," https:// ww:whitehouse:gov/issues/education/highereducation/ making-college-affordable, 2016.

[2]. Complete College America, "Four-year myth: Making college more affordable," http:// completecollege:org/ wpcontent/uploads/2014/ 11/4-Year-Myth:pdf, 2014.

[3]. H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis–a general method for cognitive model evaluation and improvement," in International Conference on Intelligent Tutoring Systems. Springer, 2006, pp. 164–175.

[4] M. Feng, N. Heffernan, and K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," User Modeling and User-Adapted Interaction, vol. 19, no. 3, pp. 243–266, 2009.

[5] H.-F. Yu, H.-Y. Lo, H.-P. Hsieh, J.-K. Lou, T. G. McKenzie, J.-W. Chou, P.-H. Chung, C.-H. Ho, C.-F. Chang, Y.-H. Wei et al., "Feature engineering and classifier ensemble for kdd cup 2010," in Proceedings of the KDD Cup 2010 Workshop, 2010, pp. 1–16.

[6]. Z. A. Pardos and N. T. Heffernan, "Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset," Journal of Machine Learning Research W & CP, 2010.

[7]. Y. Meier, J. Xu, O. Atan, and M. van der Schaar, "Personalized grade prediction: A data mining approach," in Data Mining (ICDM), 2015 IEEE International Conference on. IEEE, 2015, pp. 907–912.

[8]. C. G. Brinton and M. Chiang, "Mooc performance prediction via clickstream data and social learning networks," in 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015, pp. 2299– 2307.

[9]. Y. Jiang, R. S. Baker, L. Paquette, M. San Pedro, and N. T. Heffernan, "Learning, moment-by-moment and over the long term," in International Conference on Artificial Intelligence in Education. Springer, 2015, pp. 654–657.

[10] C. Marquez-Vera, C. Romero, and S. Ventura, "Predicting school failure using data mining," in Educational Data Mining 2011, 2010.

[11]. Y.-h. Wang and H.-C. Liao, "Data mining for adaptive learning in a tesl-based e-learning system," Expert Systems with Applications, vol. 38, no. 6, pp. 6480–6485, 2011.

[12]I. N. Thai-Nghe, L. Drumond, T. Horvath, L. Schmidt-Thieme et al., "Multi-relational factorization models for predicting student performance," in Proc. of the KDD Workshop on Knowledge Discovery in Educational Data. Citeseer, 2011.

[13]. A. Toscher and M. Jahrer, "Collaborative filtering applied to educational data mining," KDD cup, 2010.

[14]. R. Bekele and W. Menzel, "A bayesian approach to predict performance of a student (bapps): A case with ethiopian students," algorithms, vol. 22, no. 23, p. 24, 2005.

[15]. N. Thai-Nghe, T. Horvath, and L. SchmidtThieme, "Factorization mod- ´ els for forecasting student performance," in Educational Data Mining 2011, 2010.

[16] A. Blaise, M. Bouet, V. Conan, and S. Secci, ‘‘Botnet fingerprinting: A frequency distributions scheme for lightweight bot detection,’’ IEEE Trans. Netw. Service Manage., vol. 17, no. 3, pp. 1701–1714, Sep. 2020.

[17] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, ‘‘A novel graph-based approach for IoT botnet detection,’’ Int. J. Inf. Secur., vol. 19, no. 5, pp. 567–577, Oct. 2020

[18] W. Yassin, R. Abdullah, M. F. Abdollah, Z. Mas’ud, and F. A. Bakhari, ‘‘An IoT botnet prediction model using frequency based dependency graph: Proof-of-concept,’’ in Proc. 7th Int. Conf. Inf. Technol., IoT Smart City, Dec. 2019, pp. 344–352