JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DATA POISONING ATTACKS ON FEDERATED MACHINE LEARNING

M. Sai Bala, MCA Student, Department of CSE, Amrita Sai institute of science and Technology, Andhra Pradesh, India
Ch. Sowjanya, Department of CSE, Amrita Sai institute of science and Technology, Andhra Pradesh, India

Abstract - Cybercrime is a growing problem worldwide, taking advantage of vulnerabilities in computer systems. Ethical hackers are increasingly focused on identifying these weaknesses and suggesting ways to protect against them. The cybersecurity community has a pressing need for the development of effective techniques to combat cyber threats. Many of the methods currently used in Intrusion Detection Systems (IDS) struggle to cope with the dynamic and complex nature of cyberattacks on computer networks. In response, the use of machine learning in cybersecurity has gained significant importance due to its effectiveness in addressing these issues. Machine learning techniques have been applied to tackle major challenges in cybersecurity, including intrusion detection, malware classification, spam detection, and phishing prevention. While machine learning cannot fully automate a cybersecurity system, it plays a crucial role in identifying threats more efficiently than traditional software-based approaches. This, in turn, eases the workload on security analysts. By employing adaptive methods, such as various machine learning techniques, we can achieve higher detection rates, lower false alarm rates, and reasonable computational and communication costs. Our primary objective is to recognize that the task of identifying cyberattacks is fundamentally distinct from other applications, which presents unique challenges. This distinctiveness makes it more challenging for the intrusion detection community to effectively implement machine learning methods..

Index Terms— Cyber-crime, Machine learning, Cyber-security, Intrusion detection system

1. INTRODUCTION

In today's world, we're witnessing a rise in cyberwarfare activities carried out by political and commercial entities. Their intent is to disrupt, damage, or censor information on computer networks. When designing network protocols, it's crucial to ensure they can withstand intrusions by powerful attackers who might have control over a portion of the network. These controlled parties are capable of executing both passive (e.g., eavesdropping, nonparticipation) and active attacks (e.g., jamming, message dropping, corruption, and forgery).

To counter these threats, we employ intrusion detection, which involves continuously monitoring the events taking place in a computer system or network. The goal is to analyze these events for any signs of potential incidents and to prevent unauthorized access. This process often entails automatically collecting data from various systems and network sources and then scrutinizing this data for possible security issues.

Conventional intrusion detection and prevention methods, such as firewalls, access controls, and encryption, have their limitations in safeguarding networks and systems against increasingly

sophisticated attacks, including denial of service. Many of these systems suffer from high rates of false positives and false negatives and struggle to adapt to evolving malicious behaviors.

In the past decade, there has been a growing interest in applying Machine Learning (ML) techniques to intrusion detection. These techniques aim to enhance detection rates and adaptability. They are instrumental in keeping intrusion knowledge bases up-to-date and comprehensive.

In recent times, the issue of cybersecurity and protecting against a wide range of cyberattacks has become a pressing concern. This is primarily due to the substantial expansion of computer networks and the proliferation of applications used by individuals and organizations, especially with the widespread adoption of the Internet of Things (IoT). Cyberattacks inflict significant damage and financial losses in large-scale networks.

Traditional solutions, like hardware and software firewalls, user authentication, and data encryption, are proving inadequate to meet the challenges posed by the growing demand and the evolving landscape of cyber threats. These conventional security measures are insufficient to counter the rapidly advancing intrusion systems. Firewalls, for example, primarily control external access to networks but often fail to signal internal attacks. Hence, there is a clear need to develop effective defense techniques, such as machine learning-based Intrusion Detection Systems (IDS), to enhance system security.

In essence, an Intrusion Detection System (IDS) is a system or software that identifies malicious activities and policy violations within a network or system. It detects anomalies and abnormal behaviors during daily network operations and is instrumental in identifying security risks like denial-of-service (DoS) attacks. Additionally, an IDS assists in identifying, assessing, and mitigating unauthorized system behaviors, such as unauthorized access, modification, and destruction. There are various types of intrusion detection systems, including host-based and network-based IDS, catering to different perspectives and requirements..

2. LITERATURE SURVEY

An Intrusion Detection System (IDS) often faces several challenges, including dealing with the high volume of network traffic, highly uneven data distribution, the difficulty of establishing clear boundaries between normal and abnormal behaviors, and the need for continuous adaptation in an ever-changing environment.

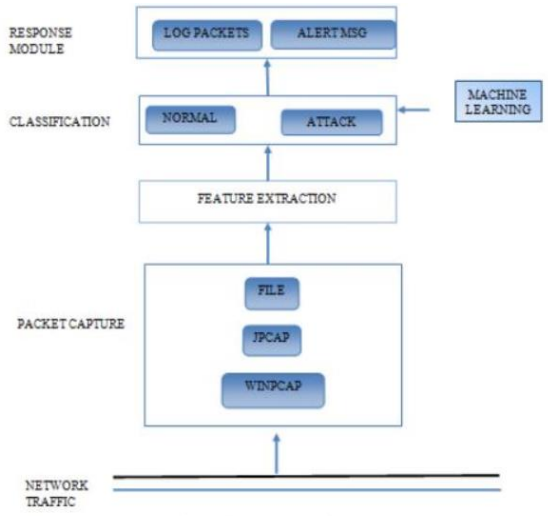
The central challenge is to efficiently capture and categorize the diverse behaviors within a computer network. Strategies for classifying network behaviors typically fall into two categories: misuse detection and anomaly detection. Misuse detection techniques scrutinize both network and system activities to identify known instances of misuse, relying on signature matching algorithms. This approach is effective at identifying known attacks. However, it often misses novel attacks, leading to false negatives. While IDS may generate alerts, reacting to every alert consumes time and resources, potentially destabilizing the system.

To address this challenge, IDS should exercise patience and not initiate the elimination process immediately upon detecting the first sign of trouble. Instead, it should accumulate alerts and make decisions based on their correlations.

Several noteworthy statistics regarding the impact of cybersecurity on businesses, organizations, and individuals include:

In recent years, cybercrime has resulted in the theft of over $400 billion and substantial costs associated with mitigating the damage caused by these crimes.
By 2022, it is expected that there will be a shortage of more than 1.8 million cybersecurity professionals.
Global organizations are projected to spend a minimum of $100 billion annually on cybersecurity protection.
Attackers currently generate over $1 billion in annual revenue from ransomware attacks, such as Wannacry and Crypto Wall attacks..

3. SYSTEM ANALYSIS:



System Architecture

EXISTING SYSTEM:

In the rapidly expanding realm of cybersecurity, it's challenging to quantify or justify why cybersecurity holds such a significant impact. Allowing malicious threats to operate without restraint, at any time or in any context, is far from acceptable and can lead to severe harm. This is particularly true within the complex web of consumers and entities using the internet and corporate data, which cybersecurity teams find increasingly difficult to protect and contain. Cybersecurity is essential for individuals, families, businesses, governments, and educational institutions operating within the global network or the internet. With the power of Machine Learning, we have the potential to enhance cybersecurity.

In today's high-tech infrastructure, which incorporates network and cybersecurity systems, an immense amount of data and analytics are collected concerning nearly all vital aspects of mission-critical systems. Nevertheless, human oversight remains a pivotal component, providing intelligent insights into contemporary infrastructure. Most intrusion detection systems primarily focus on threats to the perimeter attack surface, starting with the firewall. While this approach protects north-south traffic within your network, it often overlooks the lateral spread (east-west) that many modern network threats exploit as they infiltrate an organization's network, remaining undetected.

Research has revealed that only 20% of identified threats originate from north-south monitoring. When an intrusion detection system identifies suspicious activity, it typically reports the violation to a security information and event management (SIEM) system, where genuine threats are distinguished from benign traffic irregularities or false alarms. However, the longer it takes to recognize a threat, the more damage it can inflict. While an intrusion detection system is immensely valuable for network monitoring, its effectiveness largely depends on how the provided information is managed. Since detection tools do not prevent or resolve potential issues, they only add a security layer if there is appropriate personnel and policies to administer them and respond to threats. Intrusion detection systems cannot inspect encrypted packets, allowing intruders to penetrate the network. These intrusions are not registered by an intrusion detection system until they have penetrated deeper into the network, leaving systems vulnerable until the intrusion is discovered. This is a significant concern as encryption is increasingly used to secure data. An intrusion detection system often generates numerous false positives, in some cases outnumbering actual threats. While an intrusion detection system can be adjusted to reduce false positives, it still necessitates engineers to respond to them. Neglecting to monitor false positives can lead to genuine attacks slipping through or being overlooked.

PROPOSED SYSTEM:

Machine learning algorithms can be harnessed to train and detect cyberattacks promptly. Once an attack is identified, email notifications can be dispatched to security engineers or users. Various classification algorithms, such as Support Vector Machine (SVM), a supervised learning method that analyzes data and identifies patterns, can be employed to determine if it is a DoS/DDoS attack or not. Given that we cannot predict when, where, or how an attack may occur, and absolute prevention against such attacks cannot be guaranteed, our best current approach is early detection to mitigate the risk of irreparable damage caused by such incidents. Organizations have the option to utilize existing solutions or develop their own to identify cyberattacks at an early stage, thus minimizing their impact. Ideally, such a system should require minimal human intervention.

4. OUTPUT RESULTS:

Internet of Things enabled cyber physical systems such as Industrial equipment’s and operational IT to

send and receive data over internet. This equipment's will have sensors to sense equipment condition and report to centralized server using internet connection. Sometime some malicious users may attack or hack such sensors and then alter their data and this false data will be report to centralized server and false action will be taken. Due to false data many countries equipment got failed and many algorithms was developed to detect attack but all this algorithms suffers from data imbalance (one class my contains huge records (for example NORMAL records and other class like attack may contains few records which lead to imbalance problem and detection algorithms may failed to predict accurately). To deal with data imbalance existing algorithms were using OVER and UNDER sampling which will generate new records for FEWER class but this technique improve accuracy but not up to the mark.

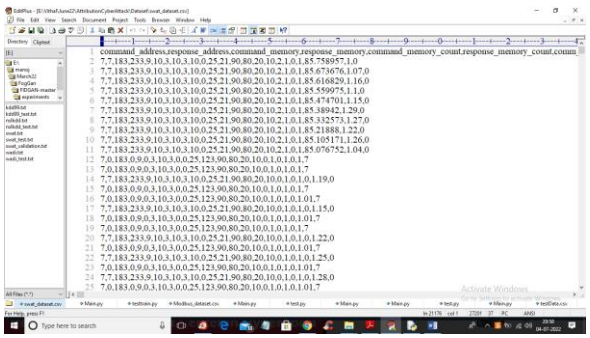
To overcome from this issue author is introducing novel technique without using any under or oversampling algorithms and this technique consists of two parts

1) Auto Encoder: auto encoder deep learning will get trained on imbalanced dataset and then extract features from it and this extracted featured will get trained with DECISION TREE algorithm to predict label for known or unknown attacks. Decision tree get trained on reduced number of features obtained from PCA (principal component analysis) algorithm.
2) Deep Neural Network (DNN): in this level DNN algorithm get trained on known and unknown attacks. If any records contains attack signature then DNN will identify attack label or class and attribute them.

To implement this project author has used SWAT (secure water treatment) and this dataset contains IOT request and response signature and associate each dataset with unique attack label and dataset contains below cyber-attack labels

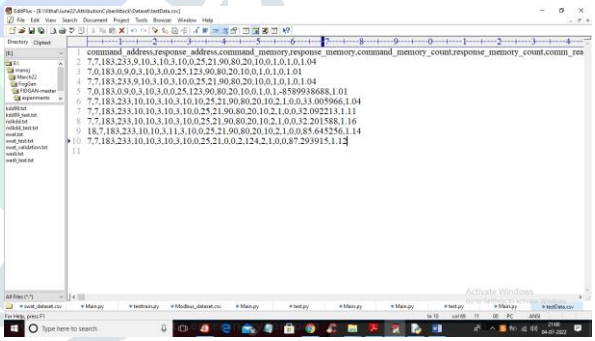
'Normal', 'Naive Malicious Response Injection (NMRI)', 'Complex Malicious', 'Response Injection (CMRI)', 'Malicious State Command Injection (MSCI)', 'Malicious Parameter Command Injection (MPCI)', 'Malicious Function Code Injection (MFCI)', 'Denial of Service (DoS)'

Above are the attacks found in dataset and dataset contains above labels as integer value of its index for example NORMAL label index will be 0 and continues up to 8 class labels. Below screen showing dataset details



In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in last column we have attack type from label 0 to 7. We will used above dataset to train propose Auto Encoder, decision tree and DNN algorithms.

In below screen we are using NEW test data which contains only signature and there is no class label and propose algorithm will detect and attribute class labels.



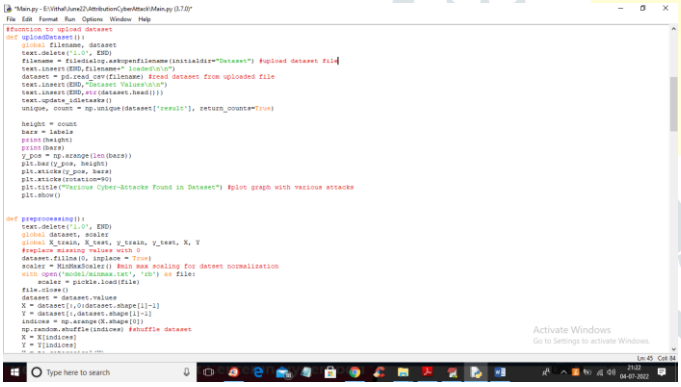
In above test data we have IOT request signature without class labels.

To implement this project we have designed following modules

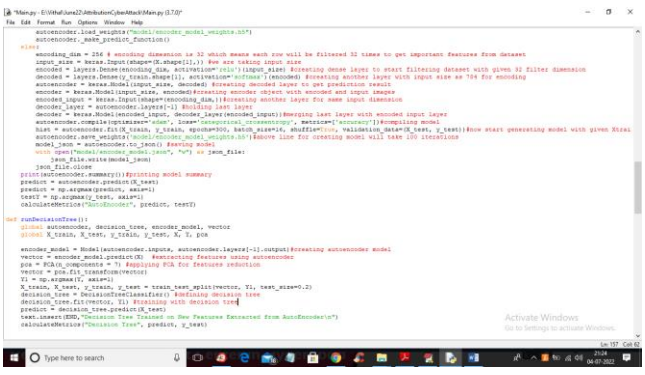
1) Upload SWAT Water Dataset: using this module we will upload dataset to application and then read dataset and then find different attacks found in dataset
2) Preprocess Dataset: using this module we will replace all missing values with 0 and then apply MIN-MAX scaling algorithm to normalized features values and then split dataset into train and test where application used 80% dataset for training and 20% for testing

3) Run AutoEncoder Algorithm: using this module we will trained AutoEncoder deep learning algorithm and then extract features from that model.
4) Run Decision Tree with PCA: extracted features from AutoEncoder will get transform using PCA to reduce features size and then retrain with Decision tree. Decision tree will predict label for each record based on dataset signatures
5) Run DNN Algorithm: predicted decision tree label will further train with DNN (deep neural network) algorithm to detect and attribute attacks
6) Detection & Attribute Attack Type: using this module we will upload unknown or un-label TEST DATA and then DNN will predict attack type
7) Comparison Graph: using this module we will plot comparison graph between all algorithms
8) Comparison Table: using this module we will display comparison table of all algorithms which contains metrics like accuracy, precision, recall and FSCORE.

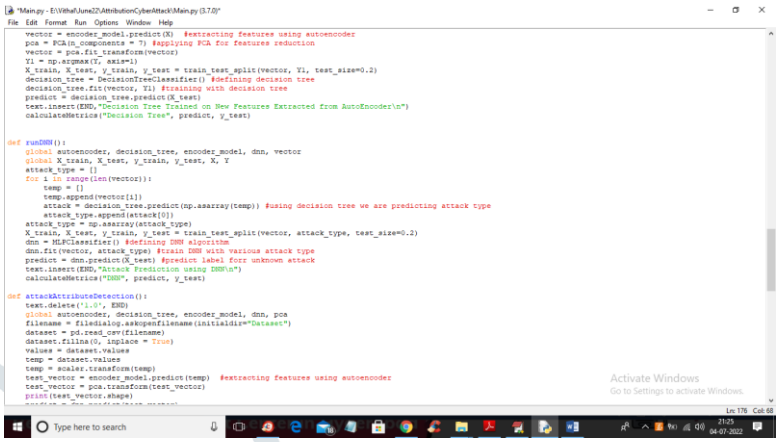
In below screen you can read red colour comments to know about algorithms implementation



In above screen read red colour comments to know about dataset loading and min-max normalization



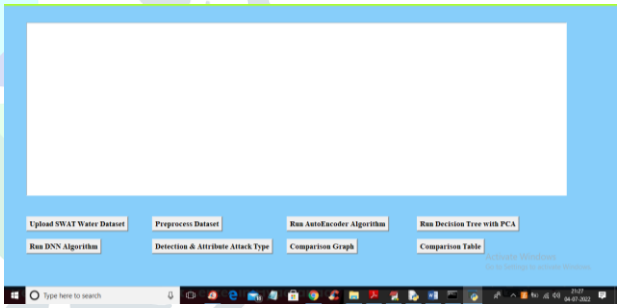
In above screen you can see we are using AutoEncoder, PCA and decision tree to train dataset and in below screen we are using DNN algorithms to train dataset with predicted labels from Decision Tree.



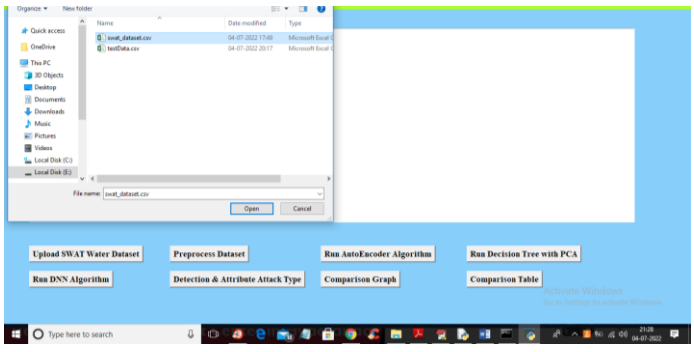
In above screen we are training dataset with DNN algorithms

SCREEN SHOTS

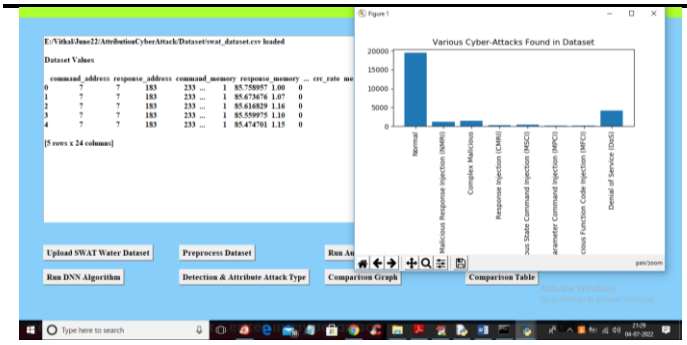
To run project double click on 'run.bat' file to get below screen



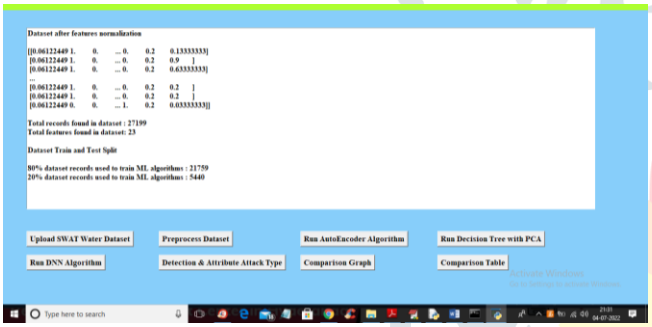
In above screen click on 'Upload SWAT Water Dataset' button to upload dataset to application and get below output



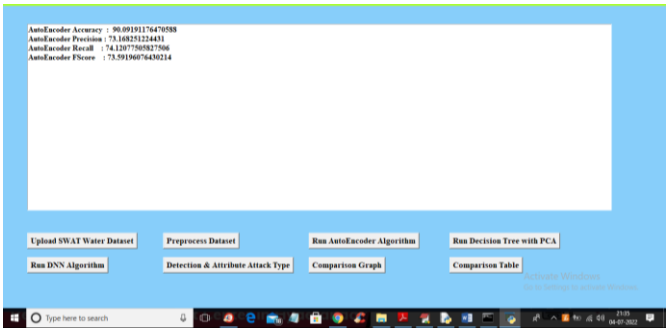
In above screen selecting and uploading SWAT dataset file and then click on 'Open' button to load dataset and get below output



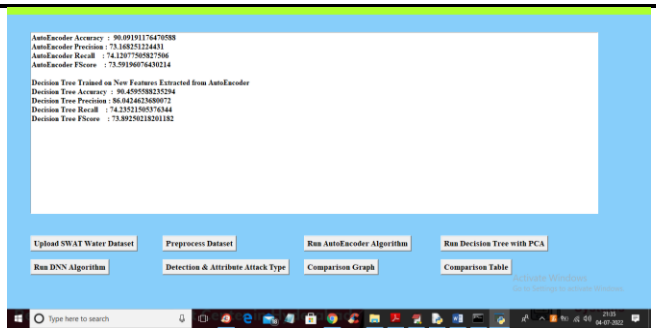
In above screen dataset loaded and in graph x-axis contains ATTACK NAME and y-axis contains count of those attacks found in dataset and we can see 'NORMAL' class contains so many records and other attacks contains very few records so it will raise data imbalance problem which can be solved using AutoEncoder, Decision Tree and DNN. Now close above graph and then click on 'Preprocess Dataset' button to remove missing values and then normalized values with MIN-MAX algorithm



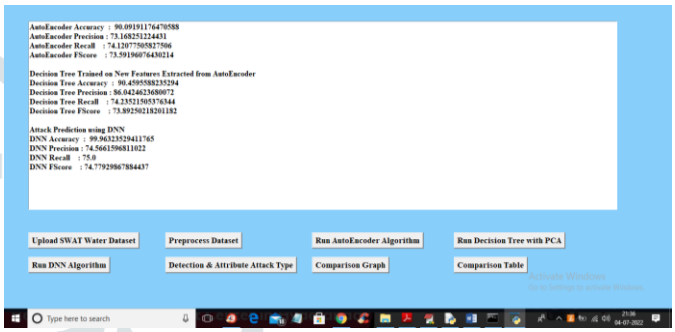
In above screen all values are normalized (converting data between 0 and 1 called as normalization) and then we can see total records in dataset and then dataset train and test split records count also displaying. Now dataset is ready and now click on 'Run AutoEncoder Algorithm' button to train dataset with AutoEncoder and get below accuracy



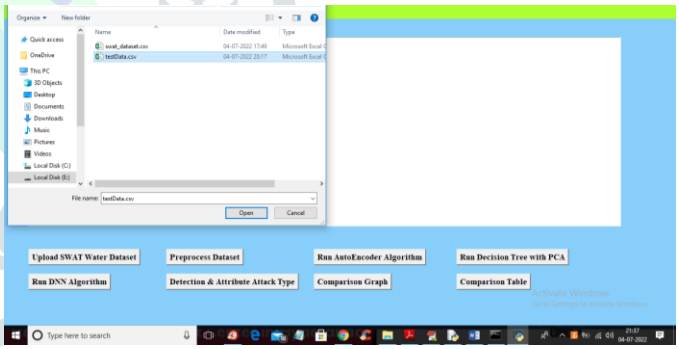
In above screen with AutoEncoder we got 90% accuracy and this accuracy can be enhance by implementing Decision Tree with PCA algorithm and now click on 'Run Decision Tree with PCA' button to get below output



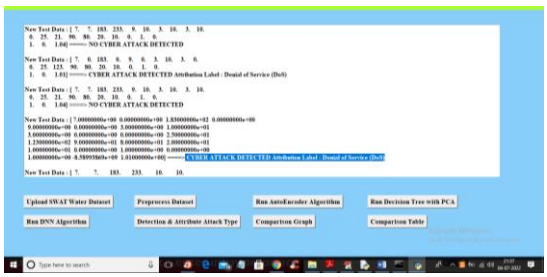
In above screen we can see with decision tree accuracy and precision value is enhanced and now click on 'Run DNN Algorithm' button to further enhance accuracy and get below output



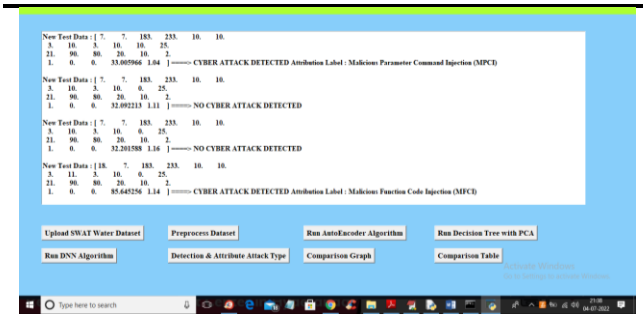
In above screen with DNN we got 99% accuracy and now click on 'Detection & Attribute Attack Type' button to upload test DATA and detect attack attributes



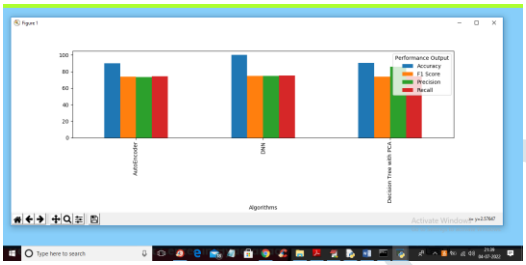
In above screen selecting and uploading 'TEST DATA' file and then click on 'Open' button to get below output



In above screen in square bracket we can see TEST data values and after arrow =➔ symbol we can see detected ATTACK TYPE and scroll down above text area to view all detection



In above screen we can see detected various attacks and now click on 'Comparison Graph' button to get below graph



In above graph x-axis represents algorithms names and y-axis represents different metric values such as precision, recall, accuracy and FSCORE with different colour bars and in all algorithms DNN got high accuracy and now close above graph and then click on 'Comparison Table' to get below comparison table of all algorithms

Algorithm Name	Accuracy	Precision	Recall	FSCORE
AutoEncoder	90.09191176470588	73.16825122244431	74.12077505827506	73.59196076430214
Decision Tree with PCA	90.45955882353394	86.04346236800727	74.23521505376344	73.89250211201182
DNN	99.96323529411765	74.56615968811022	75.0	74.77929867884437

In above table we can see algorithm names and its metrics values such as accuracy and precision and other.

5. CONCLUSION

This article proposed the utilization of Artificial Intelligence (AI) for the identification of application layer attacks. It employed a graph-based segmentation method and dynamic programming to extract instances, represented as PCRE standard expressions, for the model. These standard expressions were utilized as a reference to exhibit the real behavior of applications and identify digital attacks effectively. Furthermore, the study presented findings demonstrating the successful application of the proposed algorithm in locating application layer attacks.

REFERENCES

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

[2] Complete College America, "Four-year myth: Making college more affordable," http://completecollege:org/wp-content/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] KDD Cup, "Educational data minding challenge," https://pslcdatashop: web:cmu:edu/KDDCup/, 2010.

[10] 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.

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

[12] 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.

[13] 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.

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

[15] 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.