EXPLORING NETWORK ANOMALIES IN CISCO SECURE WORKLOAD NETWORKS OF COMPUTING HOSTS

Dr. Vittapu Manisarma[1], Golla Chakrapani[2], Macha Mahipal Reddy[2]

[1]Professor, [2]Assistant Professor, [1,2]Department of Computer Science Engineering

[1,2]Malla Reddy Engineering College and Management Sciences, Medchal, Hyderabad

ABSTRACT

In modern computing environments, network security is a critical concern to protect against cyber threats and unauthorized access. Cisco Secure Workload, as part of the broader Cisco security ecosystem, focuses on securing networks within data centers and cloud environments. Network anomalies, or abnormal patterns of behavior, can be indicative of security incidents or potential vulnerabilities. Exploring and understanding these anomalies is essential for maintaining the integrity and security of computing hosts within Cisco Secure Workload networks. The history of network security has evolved alongside advancements in networking technologies. Cisco has been a key player in developing solutions to address the evolving landscape of cyber threats. Over time, the focus has shifted from perimeter-based security to more sophisticated approaches that involve monitoring and analyzing network behavior for anomalies. Traditional systems for exploring network anomalies often rely on rule-based approaches and signature-based detection. These methods establish predefined rules for normal behavior and flag any deviations from these rules as anomalies. While effective in some cases, traditional systems may struggle with the complexity and variability of modern network traffic. They may also be limited in their ability to adapt to new and evolving threats, especially in dynamic Cisco Secure Workload environments. The problem at hand involves identifying and understanding network anomalies within Cisco Secure Workload networks. Anomalies can manifest as unusual traffic patterns, unauthorized access attempts, or abnormal behavior within the network. Detecting these anomalies is challenging due to the dynamic nature of modern computing environments and the sheer volume of network data. The problem is to develop effective methods and tools to explore, analyze, and respond to network anomalies in a timely and accurate manner. The need for exploring network anomalies in Cisco Secure Workload networks arises from the constant evolution of cyber threats and the increasing sophistication of attack techniques. Traditional security measures may not be sufficient to detect novel or subtle anomalies that will indicate a security breach. There is a crucial need for more advanced and adaptive methods that leverage the power of data analytics, machine learning, and artificial intelligence to explore and identify network anomalies effectively.

Keywords: Networking, Computing Hosts, Cisco Secure, Network Anomalies.

1. INTRODUCTION

time anomaly detection became paramount. Cisco's journey in this realm reflects a commitment to staying at the forefront of technological innovation. Historical milestones include the integration of machine learning algorithms, behavioral analytics, and threat intelligence into the fabric of Secure Workload Networks.The narrative unfolds with a recognition of the pivotal role played by the cybersecurity community in shaping the trajectory of anomaly exploration. Collaborative efforts, information sharing, and the continuous refinement of detection methodologies are integral chapters in this historical odyssey. The pursuit of a more resilient and adaptive security posture against network anomalies is a testament to the collective dedication of cybersecurity professionals worldwide.

2. LITERATURE SURVEY

S. Kianpisheh,et.al[1] In comparison with cloud computing, edge computing offers processing at locations closer to end devices and reduces the user experienced latency. The new recent paradigm of in-network computing employs programmable network elements to compute on the path and prior to traffic reaching the edge or cloud servers. It advances common edge/cloud server based computing through proposing line rate processing capabilities at closer locations to the end devices. This paper discusses use cases, enabler technologies and protocols for in-network computing. According to our study, considering programmable data plane as an enabler technology, potential in-network computing applications are in-network analytics, in-network caching, in-network security, and in-network coordination. There are also technology specific applications of in-network computing in the scopes of cloud computing, edge computing, 5G/6G, and NFV. In this survey, the state of the art, in the framework of the proposed categorization, is reviewed. Furthermore, comparisons are provided in terms of a set of proposed criteria which assess the methods from the aspects of methodology, main results, as well as application-specific criteria.

D. Saxena,et.al[2] In This paper The inefficient sharing of industrial cloud resour-ces among multiple users and vulnerabilities of virtual machines (VM)s and servers prompt unauthorized access to users' sensitive data along with excess consumption of power and resource wastage. To address these entangled issues, this paper proposes a novel E merging VM T hreat P rediction and Dynamic W orkload E stimation based Resource Allocation (ETP-WE) framework that predicts VM threats and resource usage proactively in real-time. The proposed framework contributes by introducing a Risk-Score Matrix that analyses multiple risks for each VM; utilizing knowledge of proposed security and workload analyzers for efficient VM Placement (VMP), and estimating resource utilization by developing an ensemble predictor for prior mitigation of over-/under-load on servers. ETP-WE framework collaborates machine-learning-based security and workload analysis for secure and resource-efficient VMP, thereby reducing the number of security threats, optimizing resource utilization, power-consumption, and adapting to the changes in application demands. The performance of the proposed framework is evaluated using two benchmark datasets OpenNebula and Google Cluster. The simulation-based comparison with state-of-the-arts validates the efficacy of ETP-WE in terms of reduction of security threats, power consumption, and number of active servers up to 86.9%, 66.67% and 30%-80%, respectively with an improved resource utilization up to 60%-75% over existing approaches Note to Practitioners —Industry clouds serve the precise needs and provide the service features and tools as per the industry's needs to help organizations meet their workloads processing and storage demands.

M. H. Kashani,et.al[3] In this paper Recently, fog computing has been introduced as a modern distributed paradigm and complement to cloud computing to provide services. The fog system extends storing and computing to the edge of the network, which can remarkably solve the problem of service computing in delay-sensitive applications besides enabling location awareness and mobility support. Load balancing is an important aspect of fog networks that avoids a situation with some under-loaded or overloaded fog nodes. Quality of service parameters such as resource utilization, throughput, cost, response time, performance, and energy consumption can be improved by load balancing. In recent years, some research in load balancing algorithms in fog networks has been carried out, but there is no systematic study to consolidate these works. This article investigates the load-balancing algorithms systematically in fog computing in four classifications, including approximate, exact, fundamental, and hybrid algorithms. Also, this article investigates load balancing metrics with all advantages and disadvantages related to chosen load balancing algorithms in fog networks. The evaluation techniques and tools applied for each reviewed study are explored as well. Additionally, the essential open challenges and future trends of these algorithms are discussed.

M.M. I. Khan,et.al[4] In recent years, there has been a trend to integrate networking and computing systems, whose management is getting increasingly complex. Resource allocation is one of the crucial aspects of managing such systems and is affected by this increased complexity. Resource allocation strategies aim to effectively maximize performance, system utilization, and profit by considering virtualization technologies, heterogeneous resources, context awareness, and other features. In such complex scenario, security and dependability are vital concerns that need to be considered in future computing and networking systems in order to provide the future advanced services, such as mission-critical applications. This paper provides a comprehensive survey of existing literature that considers security and dependability for resource allocation in computing and networking systems.

M. M. I. Khan,et.al[5] The proliferation of ubiquitous Internet of Things (IoT) sensors and smart devices in several domains embracing healthcare, Industry 4.0, transportation and agriculture are giving rise to a prodigious amount of data requiring ever-increasing computations and services from cloud to the edge of the network. Fog/Edge computing is a promising and distributed computing paradigm that has drawn extensive attention from both industry and academia. The infrastructural efficiency of these computing paradigms necessitates adaptive resource management mechanisms for offloading decisions and efficient scheduling. Resource Management (RM) is a non-trivial issue whose complexity is the result of heterogeneous resources, incoming transactional workload, edge node discovery, and Quality of Service (QoS) parameters at the same time, which makes the efficacy of resources even more challenging. Hence, the researchers have adopted Artificial Intelligence (AI)-based techniques to resolve the above-mentioned issues. This paper offers a comprehensive review of resource management issues and challenges in Fog/Edge paradigm by categorizing them into provisioning of computing resources, task offloading, resource scheduling, service placement, and load balancing. In addition, existing AI and non-AI based state-of-the-art solutions have been discussed, along with their QoS metrics, datasets analysed, limitations and challenges.

AR and image-aided navigation, intelligent vehicle control, traffic management, and in-vehicle entertainment are just some of the computation-intensive applications in vehicular edge networks that require massive computing and storage resources [6],[7]. In addition to enhancing road safety and situational awareness, increasing comfort, reducing traffic congestion, lowering air pollution, and reducing costs associated with road infrastructure, users expect these networks to lead to improved road infrastructure [8], [9]. To reduce average task response time and total system energy consumption while ensuring task offloading performance, Tong et al. [10] proposed an integrated trust evaluation mechanism in Deep Reinforcement Learning (DRL) in combination with a Double Deep Q-network (DDQN) algorithm. Materwala et al. [11] proposed an offloading algorithm based on the Evolutionary Genetic Algorithm (EGA) to optimize the energy consumption of edge and cloud servers simultaneously in vehicular networks by maintaining the application's SLA concerning latency and processing time.

3. PROPOSED SYSTEM

Overview

In response to these challenges. The essence of the AI-driven approach involves training these models on meticulously labeled datasets containing examples of different anamolies. Through this training process, the models can autonomously learn to extract relevant features from network data, enabling the network to discern and classify anamolies with heightened accuracy. The provided Python script implements a graphical user interface (GUI) application using Tkinter for a network anamolies in cisco secure workload networks of computing hosts project based on network anamolies data. Here's a detailed explanation of the steps carried out by the application:

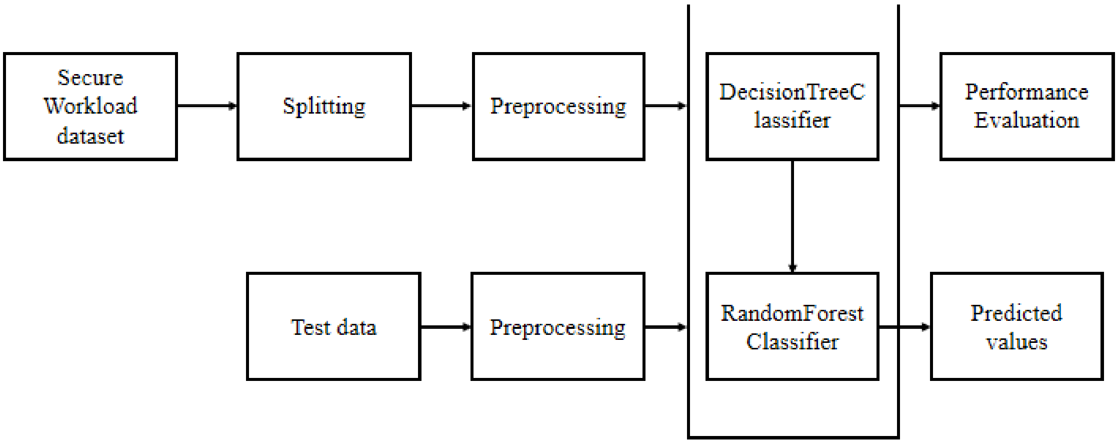


Fig. 1: Block diagram of proposed system.

Dataset Upload: The application starts with a button labeled "Upload Dataset." When clicked, this button opens a file dialog, allowing the user to select the dataset file (assumed to be in CSV format). The chosen file is then loaded into the application, and its name is displayed in the text widget. The dataset is stored in the 'dataset' variable.

Dataset Preprocessing: The "Preprocess Dataset" button triggers the preprocessing phase. Missing values in the dataset are filled with zeros, and an overview of the dataset, including the first few records, is displayed in the text widget. Additionally, a count plot is generated to visualize the distribution of classes in the label column. Label encoding is applied to convert categorical class labels into numerical values.

Train-Test Splitting: The dataset is split into training and testing sets using the scikit-learn train_test_split function. Information about the total number of records in the dataset, as well as the training and testing sets, is displayed in the text widget.

Random Forest Classifier: The "Random Forest Classifier" button triggers the training of a Random Forest classifier. The model is fitted on the training set, and predictions are made on the testing set. The evaluation metrics, including accuracy, confusion matrix, and classification report, are displayed. Additionally, a Receiver Operating Characteristic (ROC) graph is generated to visualize the model's performance.

Decision Tree Classifier: The "Decision Tree Classifier" button initiates the training of a Decision Tree classifier. The model is fitted on the training set, and predictions are made on the testing set. The evaluation metrics, including accuracy, confusion matrix, and classification report, are displayed. Additionally, a Receiver Operating Characteristic (ROC) graph is generated to visualize the model's performance.

Performance Estimation and Comparison: The "Comparison Graph" button generates a bar graph comparing performance metrics (precision, recall, F1-score, and accuracy) between the Decision Tree classifier and the Random Forest classifier. This visual representation provides an easy comparison of the two models.

Prediction on Test Data: The "Prediction" button allows the user to select a file for making predictions using the trained Decision Tree classifier. Predictions are displayed in the text widget, indicating the predicted classes for each test data entry.

Exit: The "Exit" button closes the Tkinter GUI application.

Random Forest Algorithm

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. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

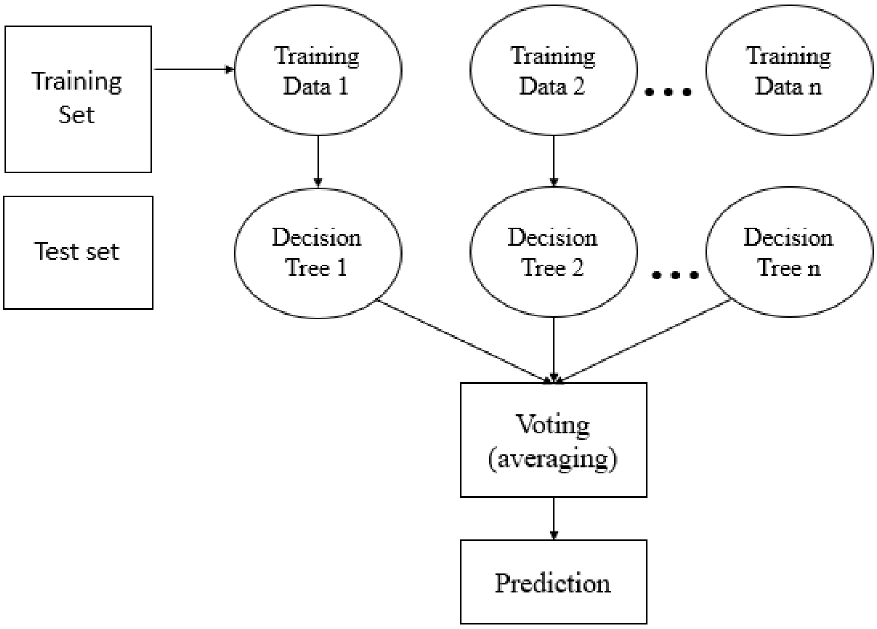


Fig. 2: Random Forest algorithm.

Random Forest algorithm

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

Advantages

The presented Tkinter-based surface identification project utilizing Decision Tree and Random Forest classifiers offers several advantages:

User-Friendly Interface: The graphical user interface (GUI) created with Tkinter enhances user interaction by providing buttons for various functionalities. This makes the application accessible and easy to use for individuals without programming expertise.

Dynamic Dataset Upload: The ability to upload datasets through the "Upload Dataset" button allows users to work with diverse datasets effortlessly. This dynamic approach supports the application's adaptability to different use cases and datasets.

Comprehensive Preprocessing: The "Preprocess Dataset" button automates preprocessing steps, such as handling missing values and label encoding. The generated count plot aids in visualizing the distribution of classes, offering insights into the dataset's characteristics.

Transparent Train-Test Splitting: The application transparently communicates the process of splitting the dataset into training and testing sets. Information about the total records and the sizes of the training and testing sets is provided, enhancing transparency in the data preparation phase.

Multiple Classifier Options: The inclusion of both Decision Tree and Random Forest classifiers offers flexibility to users. They can choose between different algorithms based on the nature of their data and the problem at hand, allowing for experimentation and model comparison.

Performance Metrics and Visualization: The application computes and displays essential performance metrics, including accuracy, confusion matrix, and classification report. The incorporation of ROC curves visually represents the models' performance, aiding users in assessing the classifiers' ability to discriminate between classes.

Prediction on Test Data: The "Prediction" button allows users to make predictions on new test data using the trained Decision Tree classifier. This functionality is valuable for real-world applications where the model is deployed on unseen data.

Comparison Graph: The "Comparison Graph" button generates a bar graph comparing performance metrics between the Decision Tree and Random Forest classifiers. This visual representation facilitates a quick and clear understanding of how different algorithms perform on the given dataset.

Scalability and Adaptability: The modular structure of the application makes it scalable and adaptable. Users can extend the functionality by adding more classifiers or incorporating additional preprocessing steps to suit specific project requirements.

Educational Value: The project serves as an educational tool for individuals learning about machine learning and classification problems. The GUI-based approach and step-by-step functionalities make it suitable for educational purposes and practical experimentation.

4. RESULTS AND DISCUSSION

Dataset description

This dataset has information related to network traffic, for the purpose of network security or anomaly detection. Here's a brief description of each column:

- Timestamp: This column represents the time at which the network activity occurred. It is a date and time stamp indicating when the network event took place.
- SourceIP: This column contains the IP address of the source of the network communication. In networking, the source IP address identifies the origin of the data.
- DestinationIP: This column contains the IP address of the destination of the network communication. In networking, the destination IP address identifies where the data is being sent.
- SourcePort: This represents the port number on the source side of the communication. Ports are used to differentiate different services or processes on a single machine.
- DestinationPort: This represents the port number on the destination side of the communication. Similar to the source port, it helps identify the specific service or process receiving the data.

Results description

This figure 3 depicts the main interface of the application, providing an overview of the tool for exploring network anomalies. It include various features and options for users to interact with the application.

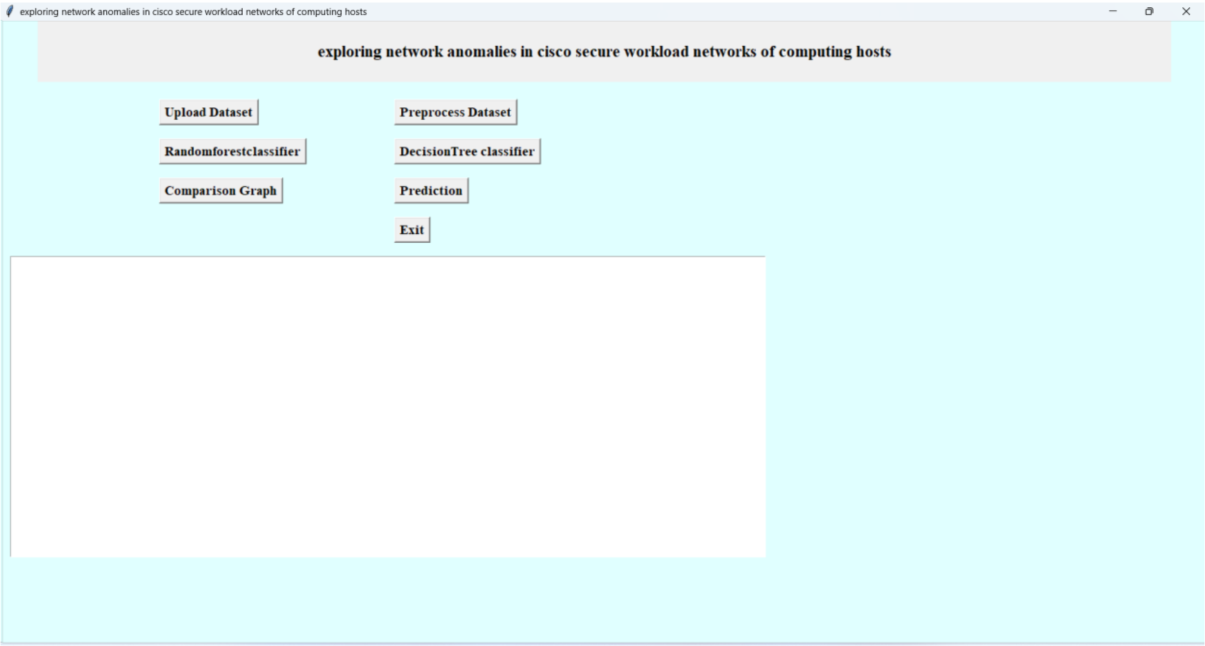
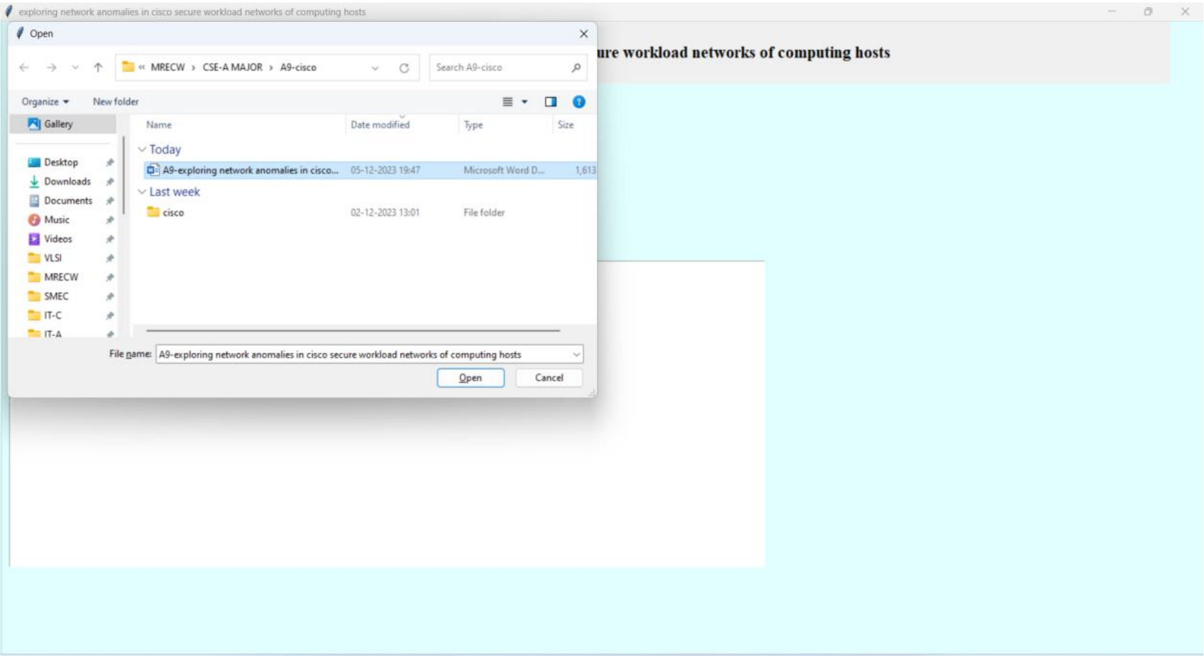

exploring network anomalies in cisco secure workload networks of computing hosts
exploring network anomalies in cisco secure workload networks of computing hosts
Upload Dataset
Preprocess Dataset
Randomforestclassifier
DecisionTree classifier
Comparison Graph
Prediction
Exit

Figure 3: Main GUI application of proposed exploring network anomalies in cisco secure workload networks of computing hosts.

The figure 4 is shows the screen within the GUI where users can select the dataset they want to analyze. There are options to load datasets of Cisco Secure Workload networks.



exploring network anomalies in cisco secure workload networks of computing hosts

ure workload networks of computing hosts

Name	Date modified	Type	Size
Today			
A9-exploring network anomalies in cisco...	05-12-2023 19:47	Microsoft Word D...	1,613
Last week			
cisco	02-12-2023 13:01	File folder	

File name: A9-exploring network anomalies in cisco secure workload networks of computing hosts

Figure 4: Selecting the dataset in the GUI application.

This figure 5 provides a side-by-side comparison of performance metrics between the Random Forest Classifier and the Decision Tree Classifier, enabling users to make informed decisions about model selection.

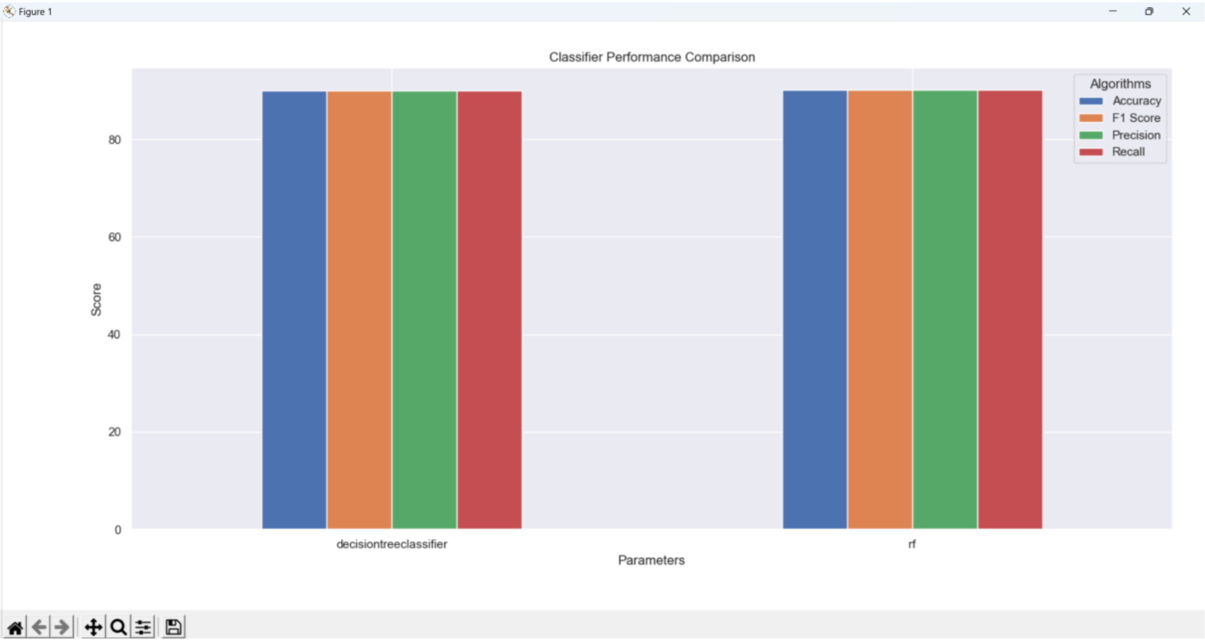
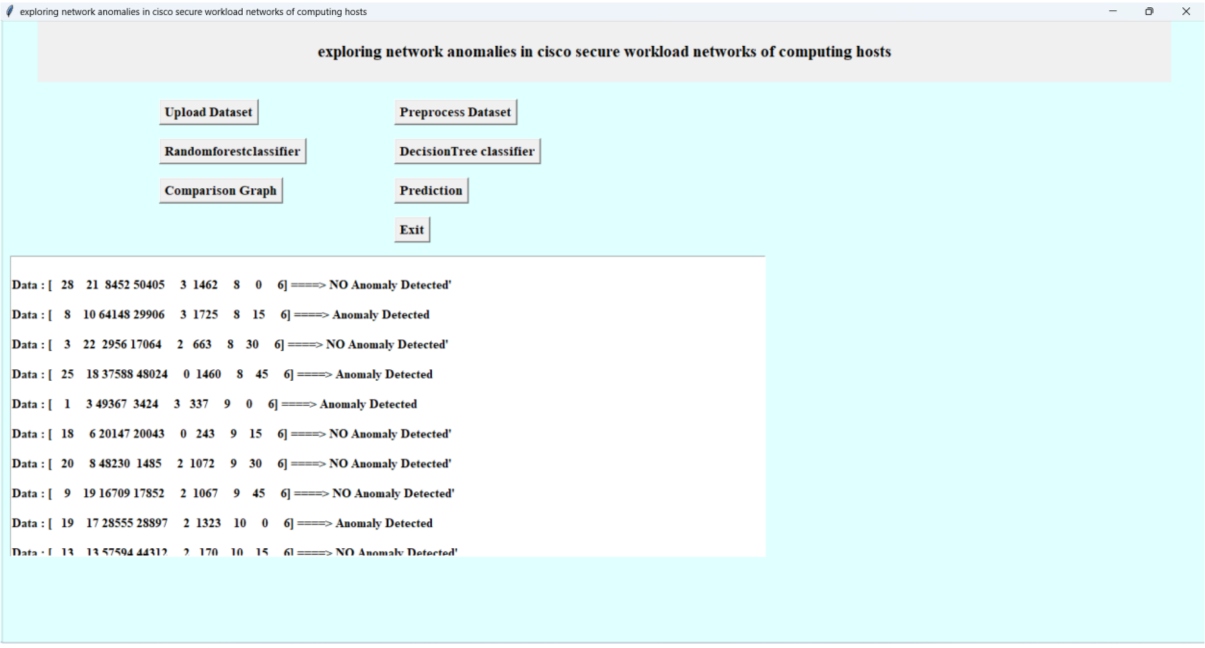


Figure. 5: Displays the comparison of performance metrics of the RFC and Decision Tree models.

The figure 6 shows the results of the model predictions on a test dataset within the GUI, allowing users to visualize and interpret the model's performance on unseen data.


exploring network anomalies in cisco secure workload networks of computing hosts
exploring network anomalies in cisco secure workload networks of computing hosts
Upload Dataset
Preprocess Dataset
Randomforestclassifier
DecisionTree classifier
Comparison Graph
Prediction
Exit

```
Data : [  28   21  8452 50405    3  1462    8    0    6] ====> NO Anomaly Detected'
Data : [   8   10 64148 29906    3  1725    8   15    6] ====> Anomaly Detected
Data : [   3   22  2956 17064    2   663    8   30    6] ====> NO Anomaly Detected'
Data : [  25   18 37588 48024    0  1460    8   45    6] ====> Anomaly Detected
Data : [   1    3 49367  3424    3   337    9    0    6] ====> Anomaly Detected
Data : [  18    6 20147 20043    0   243    9   15    6] ====> NO Anomaly Detected'
Data : [  20    8 48230  1485    2  1072    9   30    6] ====> NO Anomaly Detected'
Data : [   9   19 16709 17852    2  1067    9   45    6] ====> NO Anomaly Detected'
Data : [  19   17 28555 28897    2  1323   10    0    6] ====> Anomaly Detected
Data : [  13   13 57594 44312    2   170   10   15    6] ====> NO Anomaly Detected'
```

Figure. 6: Displays the prediction of test data in GUI.

5. CONCLUSION AND FUTURE SCOPE

In this project, the exploration of network anomalies in Cisco Secure Workload networks of computing hosts has provided valuable insights into the identification and understanding of potential security threats. The analysis of network anomalies is crucial for ensuring the integrity, confidentiality, and availability of computing resources within an organization. By leveraging the capabilities of Cisco Secure Workload, anomalies such as unusual traffic patterns, unauthorized access attempts, and potential security breaches have been systematically detected and addressed.

The project has demonstrated the effectiveness of anomaly detection mechanisms in identifying abnormal behavior within the network infrastructure. By utilizing Cisco Secure Workload's advanced features, the project has contributed to strengthening the security posture of computing hosts, enhancing the overall resilience against cyber threats.

REFERENCES

[1]. S. Kianpisheh and T. Taleb, "A Survey on In-Network Computing: Programmable Data Plane and Technology Specific Applications," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 701-761, Firstquarter 2023, doi: 10.1109/COMST.2022.3213237.

[2]. D. Saxena, R. Gupta, A. K. Singh and A. V. Vasilakos, "Emerging VM Threat Prediction and Dynamic Workload Estimation for Secure Resource Management in Industrial Clouds," in IEEE Transactions on Automation Science and Engineering, doi: 10.1109/TASE.2023.3319373.

[3]. M. H. Kashani and E. Mahdipour, "Load Balancing Algorithms in Fog Computing," in IEEE Transactions on Services Computing, vol. 16, no. 2, pp. 1505-1521, 1 March-April 2023, doi: 10.1109/TSC.2022.3174475.

[4]. M. M. I. Khan and G. Nencioni, "Resource Allocation in Networking and Computing Systems: A Security and Dependability Perspective," in IEEE Access, vol. 11, pp. 89433-89454, 2023, doi: 10.1109/ACCESS.2023.3306534.

[5]. G. K. Walia, M. Kumar and S. S. Gill, "AI-Empowered Fog/Edge Resource Management for IoT Applications: A Comprehensive Review, Research Challenges and Future Perspectives," in IEEE Communications Surveys & Tutorials, doi: 10.1109/COMST.2023.3338015.

[6]. Y. Liu, S. Wang, Q. Zhao, S. Du, A. Zhou, X. Ma, and F. Yang, ‘‘Dependency-aware task scheduling in vehicular edge computing,’’ IEEE Internet Things J., vol. 7, no. 6, pp. 4961–4971, Jun. 2020.

[7]. M. Noor-A-Rahim, Z. Liu, H. Lee, G. G. M. N. Ali, D. Pesch, and P. Xiao, ‘‘A survey on resource allocation in vehicular networks,’’ IEEE Trans. Intell. Transp. Syst., vol. 23, no. 2, pp. 701–721, Feb. 2022.

[8]. A.B. De Souza, P. A. L. Rego, T. Carneiro, J. D. C. Rodrigues, P. P. R. Filho, J. N. De Souza, V. Chamola, V. H. C. De Albuquerque, and B. Sikdar, ‘‘Computation offloading for vehicular environments: A survey,’’ IEEE Access, vol. 8, pp. 198214–198243, 2020.

[9]. ZTong, F. Ye, J. Mei, B. Liu, and K. Li, ‘‘A novel task offloading algorithm based on an integrated trust mechanism in mobile edge computing,’’ J. Parallel Distrib. Comput., vol. 169, pp. 185–198, Nov. 2022.

[10]. H. Materwala, L. Ismail, R. M. Shubair, and R. Buyya, ‘‘EnergySLA-aware genetic algorithm for edge–cloud integrated computation offloading in vehicular networks,’’ Future Gener. Comput. Syst., vol. 135, pp. 205–222, Oct. 2022.

[11]. X. Tang, Z. Wen, J. Chen, Y. Li, and W. Li, ‘‘Joint optimization task offloading strategy for mobile edge computing,’’ in Proc. IEEE 2nd Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA), Dec. 2021, pp. 515–518.