MAJOR PROJECT SYNOPSIS ON
"NETWORK ANOMALIES DETECTION"

JABALPUR ENGINEERING COLLEGE

DEPARTMENT OF
MASTER OF COMPUTER APPLICATION
(SESSION 2023-2024)

Under the Supervision of:	Under the Guidance of:
Dr. Samar Upadhyay (HOD) **Dr. Mamta Lambert (Professor)**	**Ms. Roshni Vinodia** **Ms. Anjali Sahu** **Ms. Megha Malik** **Ms. Swapnilita Kashyap** **Ms. Shikha Singh Narela**

Submitted By:

Aman Joshi - 0201CA221007

[Semester - 4]

INDEX

PREFACE

In the rapidly evolving landscape of digital connectivity, the reliance on complex network infrastructures has become ubiquitous. As organizations increasingly leverage the power of interconnected systems to drive innovation and efficiency, the vulnerability to network anomalies has emerged as a critical concern. The ability to detect and mitigate these anomalies has become paramount in ensuring the integrity, availability, and security of networked environments.

In the dynamic landscape of modern information technology, the robustness and security of computer networks have become paramount. As organizations rely increasingly on interconnected systems to facilitate their operations, the detection and mitigation of network anomalies have emerged as critical components in ensuring the integrity, availability, and confidentiality of sensitive data. This major project delves into the intricate realm of Network Anomalies Detection, offering a comprehensive exploration of methodologies, tools, and strategies to safeguard network infrastructures from evolving threats.

The exponential growth of networked devices and the ever-expanding threat landscape pose substantial challenges to the reliability of communication networks. With cyber threats becoming more sophisticated and diverse, the traditional approaches to network security prove inadequate. Consequently, the need for advanced anomaly detection mechanisms has become imperative.

This major project aims to equip readers with a profound understanding of the concepts, theories, and practical implementations associated with network anomalies detection. It delves into the nuances of anomaly detection algorithms, machine learning models, and statistical methods that form the bedrock of effective network security measures. Through a blend of theoretical discussions and hands-on examples, this project serves as a comprehensive guide for both novices and seasoned professionals seeking to fortify their networks against unforeseen and malicious activities.

The journey through this project unfolds progressively, beginning with an exploration of the foundational concepts of network anomalies and their implications. It then navigates through the intricacies of various detection techniques, highlighting their strengths, limitations, and real-world applications. Practical considerations, such as data preprocessing, feature selection, and model evaluation, are given due attention to ensure the applicability of the presented solutions in diverse network environments.

INTRODUCTION

In the dynamic landscape of modern computing, where networks serve as the backbone of information exchange, the reliable and secure operation of these networks is of paramount importance. As organizations increasingly rely on interconnected systems to facilitate communication, collaboration, and data transfer, the potential for network anomalies and security breaches has become a critical concern. Recognizing the imperative to safeguard against unforeseen disruptions and malicious activities, the focus on Network Anomalies Detection (NAD) has gained prominence.

Network anomalies encompass a wide range of irregularities within network behavior, ranging from unintentional glitches to deliberate attacks. Identifying and mitigating these anomalies is crucial to maintaining the integrity, availability, and confidentiality of networked systems. The field of Network Anomalies Detection involves the use of advanced technologies, algorithms, and analytical methods to monitor, analyze, and respond to deviations from normal network behavior.

This major project aims to delve into the intricacies of Network Anomalies Detection, exploring cutting-edge techniques and methodologies to enhance the resilience of networks against potential threats. The project will address the challenges associated with real-time anomaly detection, considering the evolving nature of network environments and the diversity of potential anomalies.

The significance of this project lies in its potential to contribute to the development of robust and adaptive systems capable of autonomously identifying and responding to network anomalies. By leveraging machine learning, statistical analysis, and anomaly detection algorithms, we seek to create a comprehensive framework for early detection and mitigation of network anomalies, thereby bolstering the overall security posture of networked infrastructures.

As we embark on this endeavor, our primary objectives include the exploration of novel anomaly detection models, the evaluation of their effectiveness in diverse network scenarios, and the development of practical solutions that can be integrated seamlessly into existing network architectures. By addressing these objectives, this major project aspires to make a meaningful contribution to the field of cybersecurity, reinforcing the resilience of networks in the face of evolving threats.

SCOPE AND OBJECTIVE OF THE SYSTEM

Scope:

The scope of the Network Anomalies Detection System is to design, implement, and deploy a robust solution capable of identifying and mitigating abnormal activities within a computer network. This system aims to cover a wide range of network anomalies, including but not limited to security threats, performance issues, and unexpected network behavior.

The key aspects of the scope include:

1. **Comprehensive Anomaly Detection:**
    - Detection of security threats such as intrusion attempts, malware, and unauthorized access.
    - Identification of performance anomalies, including bandwidth spikes, latency issues, and abnormal traffic patterns.
    - Monitoring and alerting for any deviations from normal network behavior.

2. **Real-time Monitoring:**
    - Continuous monitoring of network traffic in real-time to promptly identify anomalies as they occur.
    - Immediate alerts and notifications to network administrators upon the detection of suspicious activities.

3. **Scalability:**
    - Design the system to be scalable, ensuring it can handle the demands of networks with varying sizes and complexities.
    - Support for the integration of additional sensors and data sources to enhance anomaly detection capabilities.

4. **Adaptability:**
    - Incorporate machine learning algorithms to adapt to evolving network patterns and learn from historical data.
    - Provide mechanisms for fine-tuning and updating anomaly detection models to improve accuracy over time.

5. **User-Friendly Interface:**
    - Develop a user-friendly dashboard/interface for network administrators to visualize and interpret anomaly reports.

- Provide tools for investigation and analysis of detected anomalies, aiding in efficient response and mitigation.

6. **Compatibility:**
    - Ensure compatibility with a variety of network environments, devices, and protocols to offer a broad applicability.
    - Support interoperability with existing security systems and network infrastructure.

Objectives:

The primary objectives of the Network Anomalies Detection System are:

1. **Early Detection:**
    - Identify anomalies at the earliest stage possible to minimize potential damage and mitigate security risks promptly.
    -
2. **Accuracy:**
    - Achieve a high level of accuracy in anomaly detection to minimize false positives and negatives, ensuring reliable and actionable alerts.

3. **Responsiveness:**
    - Provide real-time alerts and notifications to enable swift response by network administrators in addressing potential threats or issues.

4. **Scalability and Performance:**
    - Design the system to handle varying network sizes and loads efficiently, ensuring optimal performance as the network scales.

5. **Continuous Improvement:**
    - Implement mechanisms for continuous learning and improvement of the anomaly detection models based on ongoing network activities and emerging threats.

6. **Interoperability:**
    - Ensure seamless integration with existing network infrastructure and security systems to enhance the overall security posture.

7. **User Empowerment:**
    - Empower network administrators with a user-friendly interface, informative dashboards, and tools for effective investigation and response to anomalies.

FEASIBILITY STUDY

1. **Introduction:**
   The purpose of this feasibility study is to assess the viability and practicality of implementing a network anomalies detection system as a major project. The project aims to enhance network security by identifying and mitigating abnormal activities within the network infrastructure.

2. **Objectives:**
   Develop a robust network anomalies detection system capable of identifying various types of security threats.
   Enhance overall network security by proactively detecting and responding to anomalies in real-time.
   Provide a user-friendly interface for monitoring and managing detected anomalies.
   Integrate the system seamlessly into existing network infrastructure.

3. **Technical Feasibility:**
   Hardware Requirements: Evaluate the existing hardware infrastructure to ensure it meets the requirements for the network anomalies detection system. Assess the need for additional servers, storage, and network equipment.

   Software Requirements: Identify the necessary software components, including anomaly detection algorithms, data analysis tools, and integration capabilities with existing network management systems.

   Scalability: Ensure that the proposed system can scale to accommodate the size and complexity of the network. Consider potential future growth and adaptability to emerging technologies.

4. **Financial Feasibility:**
   Cost Estimation: Estimate the overall cost of implementing the network anomalies detection system, including hardware, software, development, testing, and maintenance. Compare these costs with the potential benefits and savings from enhanced security.

   Return on Investment (ROI): Assess the expected ROI by considering the reduction in potential security breaches, data loss, and downtime. Evaluate the economic benefits in terms of preventing financial losses and reputational damage.

5. **Operational Feasibility:**

   User Training: Evaluate the training requirements for network administrators and security personnel to effectively use and manage the anomalies detection system.

   Integration with Existing Processes: Assess how seamlessly the system can be integrated into existing network management processes. Minimize disruption to daily operations and ensure a smooth transition.

6. **Legal and Compliance Feasibility:**

   Regulatory Compliance: Ensure that the network anomalies detection system complies with relevant data protection and privacy regulations. Address any legal considerations related to monitoring network activities.

   Ethical Considerations: Assess the ethical implications of monitoring network activities and implement safeguards to protect user privacy and confidentiality.

7. **Timeline:**

   Develop a realistic timeline for the project, considering the complexity of implementation, testing phases, and potential challenges. Ensure that the project can be completed within a reasonable timeframe.

8. **Risks and Mitigation:**

   Identify potential risks such as technical challenges, resource constraints, and resistance to change. Develop mitigation strategies to address these risks and minimize their impact on the project.

SYSTEM REQUIREMENT

Hardware Requirement:

- Minimum Intel core i5 8th Gen/Ryzen 5 3th Gen
- 128GB of Hardware Storage
- 4GB of RAM and above

Software Requirement:

- Python
- Scikit-learn
- NymPy
- Pandas
- Matplotlib
- Seaborn
- Psutil
- Scikit-plot
- Pickle
- IoT-23(Dataset Model)

TECHNOLOGIES USED:

PYTHON:

Python, a dynamically-typed and high-level programming language, is celebrated for its simplicity, readability, and versatility. Conceived by Guido van Rossum in the early '90s, Python has evolved into a popular choice among developers of diverse backgrounds. Its syntax, designed for readability, fosters an accessible learning curve for beginners and enhances collaboration in projects. Python's extensive standard library and a thriving ecosystem of third-party packages on the Python Package Index (PyPI) contribute to its widespread adoption. Known for its applicability in web development, data science, machine learning, and automation, Python has become a go-to language for solving complex problems efficiently. Whether used for scripting, building web applications with frameworks like Django, or conducting advanced analytics, Python's versatility and supportive community make it a cornerstone in the programming world.

SCIKIT-LEARN:

Scikit-learn is a robust and widely-used machine learning library for Python, offering simple and efficient tools for data analysis and modeling. Developed on top of other scientific computing libraries such as NumPy, SciPy, and Matplotlib, scikit-learn provides an extensive set of machine learning algorithms for tasks such as classification, regression, clustering, and dimensionality reduction. One of its strengths lies in its user-friendly and consistent API, making it accessible for both beginners and seasoned machine learning practitioners. Scikit-learn also supports various data preprocessing techniques, model evaluation methods, and tools for feature selection, making it a comprehensive solution for end-to-end machine learning workflows. With a strong emphasis on code simplicity and readability, scikit-learn has become an essential tool in the Python ecosystem, playing a pivotal role in the development and deployment of machine learning applications across various domains.

NUMPY:

NumPy, short for Numerical Python, is a fundamental library in the Python programming language for numerical and mathematical operations. Created by Travis Olliphant, it provides support for large, multi-dimensional arrays and matrices, along with an assortment of high-level mathematical functions to operate on these arrays. NumPy is a cornerstone in the Python data science ecosystem, serving as the foundation for many

other libraries such as Pandas, SciPy, and scikit-learn. Its efficient and optimized operations make it an essential tool for tasks involving numerical computations, data manipulation, and scientific computing. NumPy's array-oriented computing paradigm allows for concise and expressive code, making it particularly valuable for tasks like linear algebra, statistical analysis, and signal processing. Its widespread adoption has significantly contributed to Python's prominence in the field of data science and scientific computing.

PANDAS:

Pandas is a powerful open-source data manipulation and analysis library for Python. Developed by Wes McKinney and first released in 2008, Pandas provides high-performance, easy-to-use data structures—mainly Series and DataFrame—that make working with structured data seamless. It has become an indispensable tool in the field of data science and analysis. Pandas simplifies tasks such as data cleaning, transformation, exploration, and visualization, offering functionalities akin to those found in SQL and spreadsheet software. The library excels in handling heterogeneous and labeled data, allowing users to effortlessly manage and analyze datasets. Its integration with other Python libraries, such as NumPy and Matplotlib, further enhances its capabilities. Whether you are loading data from various sources, aggregating information, or performing complex data manipulations, Pandas remains a cornerstone in the Python ecosystem, empowering data scientists and analysts to efficiently work with tabular data.

MATPLOTLIB:

Matplotlib stands as a cornerstone in the Python ecosystem for data visualization. Developed by John D. Hunter in 2003, this open-source library provides a flexible and comprehensive set of tools for creating static, animated, and interactive visualizations in Python. Matplotlib's syntax and functionality are inspired by MATLAB, making it intuitive for users familiar with scientific computing environments. The library supports a wide array of plot types, including line plots, scatter plots, bar plots, histograms, and more. Matplotlib's customization options allow users to fine-tune every aspect of a plot, from colors and labels to gridlines and annotations. Additionally, it seamlessly integrates with NumPy, another essential library in the Python scientific computing ecosystem. Matplotlib's versatility, coupled with its extensive documentation and a vast community, makes it an indispensable tool for researchers, data scientists, and engineers seeking to convey insights through visualizations in Python.

SEABORN:

Seaborn is a powerful and visually appealing data visualization library built on top of Matplotlib in Python. Developed to complement Matplotlib, Seaborn provides a high-level interface for creating informative and attractive statistical graphics. Its simplicity and integration with Pandas data structures make it a popular choice for data scientists and analysts. Seaborn simplifies the process of generating complex visualizations, including heatmaps, violin plots, pair plots, and more, with minimal code. It comes with built-in themes and color palettes to enhance the aesthetics of plots. Seaborn's ability to work seamlessly with Pandas DataFrames and its focus on statistical exploration make it an excellent tool for both exploratory data analysis and communication of insights. Whether you're a beginner exploring data or an experienced data scientist, Seaborn's capabilities contribute to creating compelling and insightful visualizations with ease.

PSUTIL:

psutil is a Python cross-platform library that provides an interface for retrieving information on system utilization and managing processes. Developed to simplify system monitoring and management tasks, psutil exposes various APIs to access information related to CPU, memory, disk, network, and process-related metrics.

With psutil, developers can obtain real-time data on CPU usage, memory consumption, disk activity, and network usage. It also facilitates the retrieval of detailed information about running processes, including their status, resource usage, and more. The library abstracts the complexities of interacting with the underlying operating system, making it easy to integrate into Python applications.

Some key features of psutil include the ability to query system information, monitor system resources, and manage processes programmatically. It supports multiple operating systems, including Linux, Windows, and macOS, making it a versatile tool for system administrators, developers, and anyone involved in performance monitoring or process management.

SCIKIT-PLOT:

Scikit-plot is a Python library that provides a convenient interface for creating visualizations commonly used in machine learning and data science tasks. Built on top of the popular Matplotlib library, scikit-plot simplifies the process of generating essential plots, such as confusion matrices, ROC curves, precision-recall curves, and more. Its user-friendly API allows developers and data scientists to create informative visualizations with minimal code, making it a valuable tool for model evaluation and performance analysis. By

enhancing the interpretability of machine learning results, scikit-plot facilitates a deeper understanding of model behavior and aids in the decision-making process during the development and optimization of predictive models.

PICKLE:

Pickle is a Python module that provides a convenient way to serialize and deserialize Python objects. Serialization is the process of converting complex data types, such as lists or dictionaries, into a format that can be easily stored or transmitted. Pickle achieves this by converting Python objects into a byte stream. This serialized byte stream can be saved to a file or sent over a network. The deserialization process, or unpickling, involves reconstructing the original Python objects from the serialized byte stream. Pickle is widely used for tasks like saving and loading machine learning models, caching objects, and facilitating inter-process communication. Its simplicity and effectiveness make it a valuable tool for data persistence and sharing complex Python structures between different programs or systems.

IDE:

- VS Code
- JUIPTER
- ANACONDA

OS used for testing:

- MacOS
- Linux
- Windows

GOALS

The primary goals of network anomaly detection are to enhance the security and reliability of computer networks by identifying and responding to abnormal or suspicious activities. Here are the key objectives:

- **Threat Identification:**
  Detecting and identifying potential security threats or malicious activities within the network, such as unauthorized access, malware infections, or denial-of-service attacks.

- **Early Warning System:**
  Providing an early warning system to promptly alert administrators or automated systems about potential network anomalies, allowing for timely investigation and mitigation.

- **Incident Response:**
  Facilitating rapid and effective incident response by quickly isolating or mitigating the impact of identified anomalies, preventing further damage or unauthorized access.

- **Minimizing False Positives:**
  Striving to minimize false positives by refining detection algorithms and methodologies, ensuring that the system accurately identifies genuine anomalies while reducing unnecessary alerts.

- **Enhancing Network Visibility:**
  Improving overall network visibility by monitoring and analyzing network traffic patterns, device behavior, and communication flows to identify deviations from normal baseline behavior.

- **Protection Against Insider Threats:**
  Safeguarding the network against insider threats by monitoring user activities, access patterns, and data transfers to detect any anomalous behavior that may indicate malicious intent or compromised accounts.

- **Anomaly Profiling:**
  Developing comprehensive profiles of normal network behavior and communication patterns to establish a baseline for comparison. This allows the system to distinguish between normal and abnormal activities.

- **Adaptability and Learning:**
  Incorporating machine learning and adaptive algorithms to continuously learn and evolve with the changing nature of network activities, ensuring the detection system remains effective against emerging threats.

- **Compliance and Regulation:**
  Assisting organizations in meeting regulatory requirements and compliance standards by implementing robust network anomaly detection measures to safeguard sensitive data and infrastructure.

- **Resource Optimization:**
  Optimizing network resources by focusing on identifying and addressing critical anomalies, thereby improving the efficiency of security operations and reducing the likelihood of resource exhaustion during attacks.

- **Forensic Analysis:**
  Supporting forensic analysis by providing detailed information about detected anomalies, aiding in post-incident investigations, and contributing to the understanding of attack vectors and strategies.

- **User and Entity Behavior Analytics (UEBA):**
  Incorporating user and entity behavior analytics to analyze the behavior of both users and devices within the network, identifying deviations from normal patterns that may indicate security incidents.

BENEFITS:

Network anomaly detection offers several significant benefits in enhancing the security and operational efficiency of computer networks. Here are some key advantages:

- **Early Threat Detection:**
  One of the primary benefits is the early detection of security threats and anomalies in the network. By identifying unusual patterns or behaviors, the system can raise alerts or take preventive measures before a security incident escalates.

- **Reduced Response Time:**
  Early detection leads to quicker response times. Network anomaly detection allows security teams to respond promptly to potential threats, minimizing the impact of security incidents and reducing the likelihood of data breaches.

- **Mitigation of Insider Threats:**
  Anomaly detection helps in identifying anomalous user behavior, which is crucial for mitigating insider threats. By monitoring activities such as unauthorized access or data exfiltration, organizations can prevent malicious actions from within.

- **Optimized Resource Allocation:**
  The system allows organizations to allocate security resources more effectively. By focusing on genuine anomalies and potential threats, security teams can prioritize their efforts and resources where they are needed most.

- **Improved Network Visibility:**
  Network anomaly detection provides enhanced visibility into network traffic and activities. This increased visibility helps organizations better understand their network environment, detect unusual patterns, and proactively address potential vulnerabilities.

- **Minimization of False Positives:**
  Advanced anomaly detection systems strive to minimize false positives, ensuring that security teams are not overwhelmed with unnecessary alerts. This allows for more efficient use of human resources in investigating and responding to real threats.

- **Enhanced Compliance:**
  Many regulatory standards and compliance requirements mandate robust security measures. Network anomaly detection helps organizations meet these standards by providing proactive monitoring and threat detection capabilities.

- **Adaptability to Evolving Threats:**
  Anomaly detection systems often employ machine learning and adaptive algorithms, allowing them to evolve and adapt to new and emerging threats. This adaptability ensures that the system remains effective against a constantly changing threat landscape.

- **Forensic Analysis and Incident Investigation:**
  In the event of a security incident, anomaly detection systems contribute valuable data for forensic analysis. Detailed information about the detected anomalies helps security teams understand the nature of the attack, identify vulnerabilities, and strengthen defenses.

- **Reduction in Downtime and Business Impact:**
  Timely detection and response to anomalies contribute to minimizing downtime and business disruption. By preventing or mitigating the impact of security incidents, organizations can maintain business continuity and protect their reputation.

- **Identification of Zero-Day Attacks:**
  Anomaly detection is effective in identifying zero-day attacks or previously unknown threats. By focusing on deviations from normal behavior, the system can flag suspicious activities even if there is no prior signature or definition for the attack.

- **Continuous Monitoring and Improvement:**
  Anomaly detection systems provide continuous monitoring, allowing organizations to assess and improve their security posture over time. Regular analysis of detected anomalies contributes to refining detection algorithms and strengthening overall security.

MODULES

The data model used in network anomaly detection plays a crucial role in the accurate identification of abnormal behavior within a network. The choice of a data model depends on the specific requirements of the anomaly detection system and the nature of the network being monitored. Here are some common data models used in network anomaly detection:

- ## Flow-Based Data Model:

  Description: This model focuses on capturing and analyzing network flows, which represent the communication between different devices or hosts in the network. A flow is typically defined by source and destination IP addresses, port numbers, and the protocol used.

  Advantages: It provides a holistic view of network activity, including information about communication patterns, data volumes, and durations. Flow-based models are well-suited for detecting anomalies in network traffic.

- ## Packet-Based Data Model:

  Description: This model involves capturing and analyzing individual packets that traverse the network. It examines the content and structure of each packet to identify patterns or anomalies that may indicate malicious activity.

  Advantages: Offers granular visibility into network traffic and enables the detection of specific packet-level anomalies. It can be useful for identifying threats such as intrusion attempts or unusual payload patterns.

- ## Log-Based Data Model:

  Description: This model involves collecting and analyzing log data generated by various network devices, servers, and applications. Logs can include information about user authentication, access attempts, and other system events.

  Advantages: Allows for the correlation of events across different parts of the network, providing a comprehensive view of user and system activities. Log-based models are effective in detecting anomalies related to user behavior.

- ## Behavioral-Based Data Model:

  Description: This model focuses on establishing a baseline of normal behavior for devices, users, or entities within the network. Deviations from this baseline are

flagged as anomalies. Machine learning techniques are often applied to identify patterns indicative of abnormal behavior.

Advantages: Offers adaptability to evolving threats and can detect anomalies that may not be explicitly defined in rule-based systems. Behavioral models excel at identifying unknown or zero-day attacks.

- ## Protocol-Based Data Model:

  Description: This model focuses on analyzing network traffic based on specific communication protocols. It involves understanding the normal patterns associated with each protocol and flagging deviations as anomalies.

  Advantages: Provides protocol-specific insights, allowing for the identification of anomalies in the context of different communication standards. This model is particularly useful for detecting protocol-based attacks.

- ## Hybrid Data Model:

  Description: A hybrid approach combines multiple data models to leverage their respective strengths. For example, combining flow-based analysis with behavioral modeling can enhance the accuracy of anomaly detection.

  Advantages: Offers a more comprehensive and robust approach by leveraging the strengths of different data models. Hybrid models can provide a more nuanced understanding of network activity.

Project Lifecycle:

The life cycle of network anomaly detection involves several stages, from initial planning to ongoing refinement. Here is a general overview of the life cycle:

- ## Planning and Requirements Analysis:

  Objective Definition: Clearly define the goals and objectives of the network anomaly detection system. Identify the specific types of anomalies to be detected and the desired outcomes.

  Resource Assessment: Evaluate the resources required, including hardware, software, and personnel. Determine the budget, time frame, and scope of the detection system.

- ## Data Collection and Preprocessing:

  Data Sources Identification: Identify relevant data sources such as network logs, packet captures, and system logs. Determine the types of data needed for anomaly detection.

  Data Preprocessing: Cleanse and preprocess the collected data to remove noise, handle missing values, and format the data for analysis. This may involve normalization, aggregation, and transformation.

- ## Baseline Establishment:

  Normal Behavior Profiling: Establish a baseline of normal network behavior by analyzing historical data. This involves creating profiles of normal patterns for different network entities, including users, devices, and applications.

- ## Anomaly Detection Algorithm Implementation:

  Algorithm Selection: Choose suitable anomaly detection algorithms based on the characteristics of the network and the types of anomalies to be detected. Common approaches include statistical methods, machine learning models, and rule-based systems.

  Model Training: Train the selected algorithms using historical data to enable them to recognize normal patterns and identify anomalies.

- ## Threshold Setting and Alert Configuration:

  Threshold Definition: Set appropriate thresholds for anomaly detection based on the characteristics of the network and the desired balance between false positives and false negatives.

  Alert Configuration: Define alerting mechanisms and configurations to notify security teams or system administrators when anomalies are detected.

- ## Real-Time Monitoring:

  Continuous Monitoring: Implement real-time monitoring of network activities using the trained anomaly detection models. Continuously analyze incoming data to identify deviations from normal behavior.

  Alert Generation: Generate alerts and notifications when anomalies are detected, providing information about the nature of the anomaly, affected entities, and potential risks.

- ## Incident Response and Investigation:

  Alert Handling: Develop procedures for handling alerts, including incident response plans. Determine the appropriate actions to be taken when anomalies are identified.

  Forensic Analysis: Conduct forensic analysis on detected anomalies to understand the root cause, impact, and potential mitigation strategies.

- ## Feedback Loop and Model Refinement:

  Feedback Collection: Gather feedback from incident investigations, false positives, and system performance. Analyze the effectiveness of the detection system.

  Model Refinement: Periodically refine anomaly detection models based on the collected feedback. This may involve updating algorithms, adjusting thresholds, or incorporating new features.

- ## Documentation and Reporting:

  Documentation: Maintain comprehensive documentation of the anomaly detection system, including configuration settings, models, and incident reports.

  Reporting: Generate regular reports summarizing the performance of the anomaly detection system, including detection rates, false positive rates, and incident response metrics.

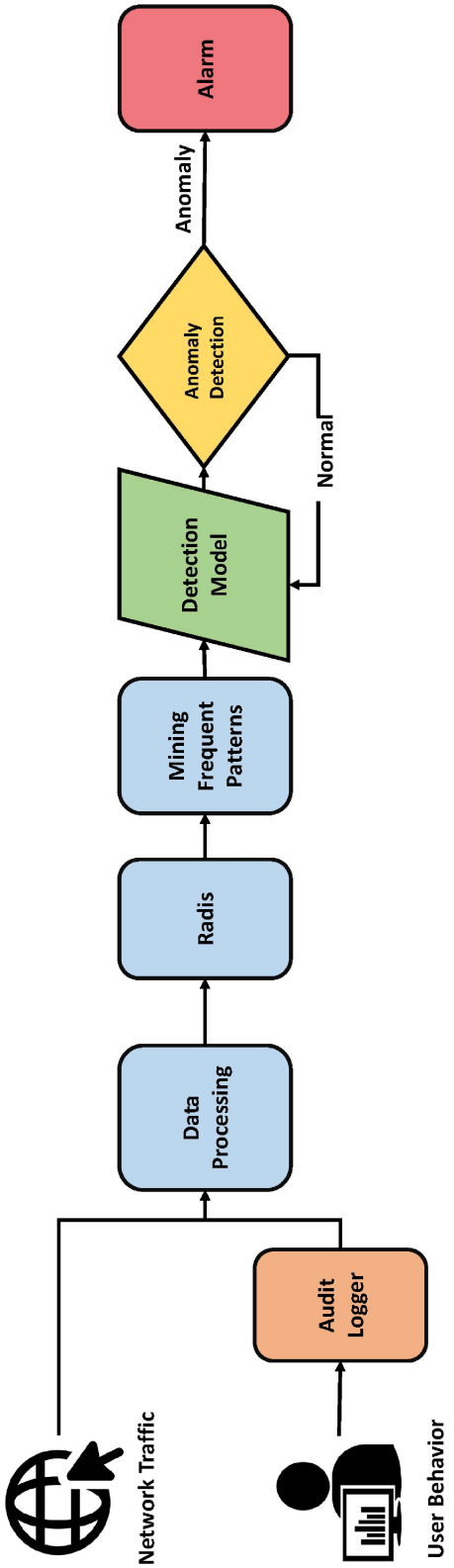
- ## Continuous Improvement:

  Adaptation to Changes: Stay vigilant to changes in the network environment, technology, and threat landscape. Adjust the anomaly detection system to adapt to evolving challenges and requirements.

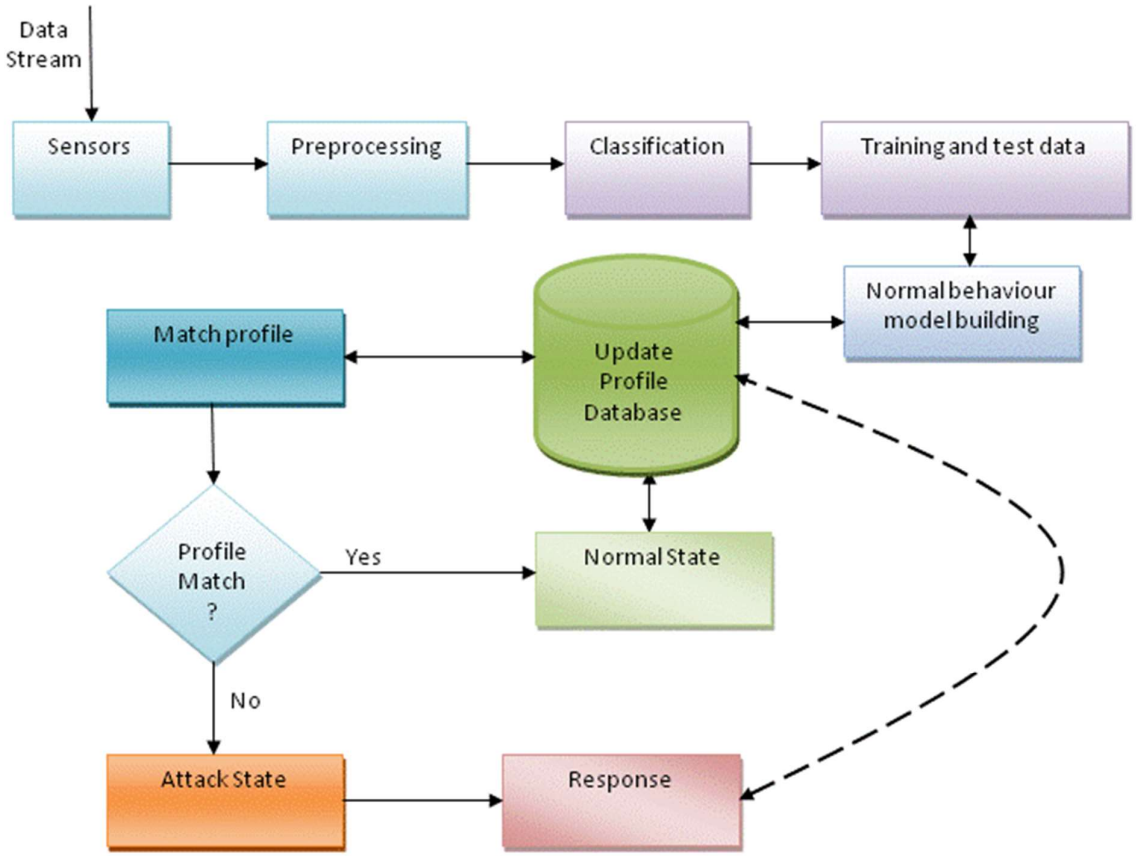
- ## Training and Awareness:

  User Training: Train security personnel on the effective use of the anomaly detection system, incident response procedures, and the interpretation of alerts.
  Awareness Programs: Conduct awareness programs to educate network users and administrators about the importance of anomaly detection and security best practices.

Data Flow Diagram:



ER-Diagram:


Data Stream
Sensors
Preprocessing
Classification
Training and test data
Normal behaviour model building
Match profile
Update Profile Database
Profile Match ?
Yes
Normal State
No
Attack State
Response

REFERENCE:

- YouTube Channels for implementation.
- Google search for research section.
- Github for open source libraries like “PICKLE”, “PSUTIL”.
- https://www.researchgate.net/profile/Ayei-Ibor/publication/282273622/figure/fig2/AS:383559685689345@1468459163382/Anomaly-Detection-Technique-for-Intrusion-Detection-Figure-2-depicts-the-anomaly.png
- https://seaborn.pydata.org/
- Dataset : https://www.stratosphereips.org/datasets-iot23
- Some Trained Models from different places.
- Some websites for the model implementation.