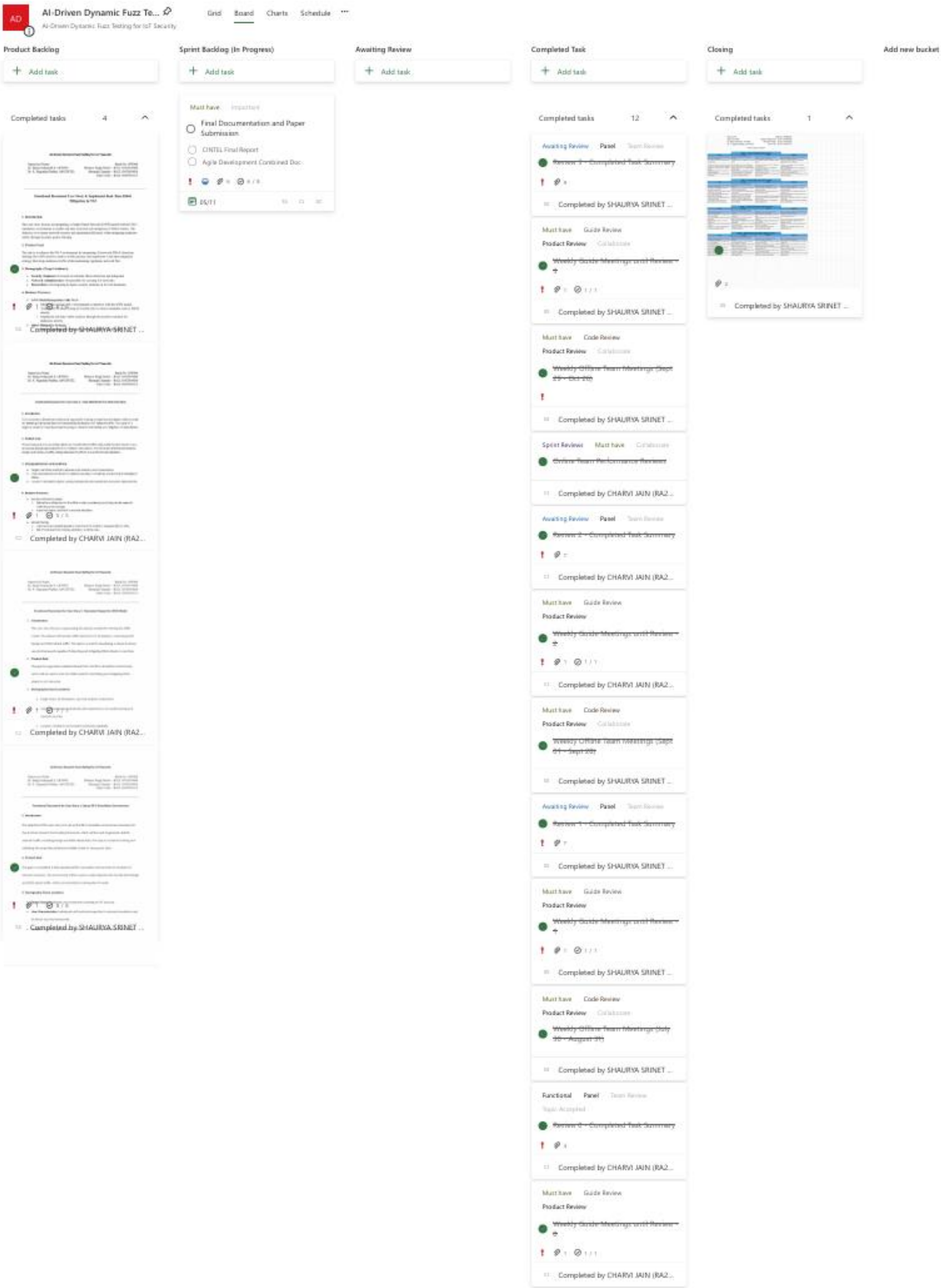
AGILE DEVELOPMENT

1. Agile MS Board



2. Product Backlog

a. User Story 1

AI-Driven Dynamic Fuzz Testing for IoT Security

~~User Story 1: Setup NS3 Simulation Environment~~

Completed on 08/31/2024 by you

SS CJ SC

Epic 1 Sprint 1 Must have

Bucket	Progress	Priority
Product Backlog	Completed	Important

Start date	Due date	Repeat
08/04/2024	08/09/2024	Does not repeat

Notes

Show on card

User Story 1: Setup NS3 Simulation Environment

- Title: Setup NS3 Simulation Environment
- Description: As a network engineer, I want to set up the NS3 simulation environment to simulate IoT network traffic, so that I can generate realistic datasets for training and validating the GNN model.
- Points: 5 points (complexity: moderate, effort: medium)
- Target Audience:
  - Network engineers responsible for configuring simulation environments.
  - Data scientists who will use the generated data for model training.
  - Researchers studying IoT network behavior.
- Estimation of Effort:
  - Environment Setup: 1 day
  - Network Configuration: 2 days
  - Simulation Testing: 1 day
  - Documentation: 1 day
  - Total Effort: 5 days
- Acceptance Criteria:
  - The NS3 environment is set up and running without errors.
  - A basic IoT network topology is configured with multiple devices and a central router.
  - The simulation generates realistic network traffic logs in the required format (XML).
  - The setup is documented, including the steps for future replication and troubleshooting.

Checklist 8 / 8

Show on card

b. User Story 2

AI-Driven Dynamic Fuzz Testing for IoT Security

~~User Story 2: Generate Dataset for GNN Training~~

Completed on 20/09/2024 by CHARVI JAIN (RA2111047010113)

SS CJ SC

Sprint 2 | Must have | Epic 2

Bucket	Progress	Priority
Product Backlog	Completed	Important
Start date	**Due date**	**Repeat**
11/08/2024	19/08/2024	Does not repeat

Notes

Show on card

- Title: Generate Dataset for GNN Training
- Description: As a data engineer, I want to generate and preprocess a dataset from the NS3 simulation logs, so that the data can be used to train the GNN model effectively.
- Points: 6 points (complexity: moderate, effort: medium)
- Target Audience:
  - Data engineers preparing the data for analysis.
  - Data scientists who will use the data for model training.
  - Cybersecurity professionals analyzing network traffic patterns.
- Estimation of Effort:
  - Data Extraction: 2 days
  - Data Cleaning & Preprocessing: 2 days
  - Feature Engineering: 4 day
  - Documentation: 1 day
  - Total Effort: 9 days
- Acceptance Criteria:
  - The dataset is successfully extracted from the NS3 logs in CSV format.
  - The dataset includes relevant features such as timestamps, source/destination IPs, packet sizes, and labels for benign or malicious traffic.
  - The dataset is balanced to prevent model bias during training.
  - The dataset is validated for consistency and completeness and is ready for training.
  - The data preprocessing steps are documented.

Checklist 7 / 7

Show on card

c. User Story 3

AI-Driven Dynamic Fuzz Testing for IoT Security

~~User Story 3: GNN Model Training for DDoS Detection~~

Completed on 20/09/2024 by CHARVI JAIN (RA2111047010113)

SS CJ SC

Sprint 3 | Epic 3 | Functional | Collaborate

Bucket	Progress	Priority
Product Backlog	Completed	Important
Start date	**Due date**	**Repeat**
21/08/2024	05/09/2024	Does not repeat

Notes

Show on card

User Story 3: Train GNN Model for DDoS Detection

- **Title:** Train GNN Model for DDoS Detection
- **Description:** As a data scientist, I want to train the GNN model using the preprocessed dataset, so that the model can accurately detect DDoS attacks in IoT network traffic.
- **Points:** 8 points (complexity: high, effort: significant)
- **Target Audience:**
  - Data scientists responsible for developing and training the AI model.
  - Cybersecurity analysts who will utilize the model for real-time threat detection.
  - AI researchers exploring the application of GNNs in network security.
- **Estimation of Effort:**
  - **Model Architecture Design:** 3 days
  - **Model Training:** 5 days
  - **Hyperparameter Tuning:** 7 days
  - **Documentation:** 1 day
  - **Total Effort:** 16 days
- **Acceptance Criteria:**
  - The GNN model is trained with an accuracy of at least 75% on the validation set.
  - The model successfully identifies and classifies DDoS traffic from benign traffic.
  - The model is tested with different hyperparameter settings to optimize performance.
  - The trained model is saved for deployment, and all training procedures are documented.

Checklist 5 / 5

Show on card

d. User Story 4

~~User Story 4: Implement Real-Time DDoS Mitigation in NS3~~

Completed on 28/09/2024 by SHAURYA SRINET (RA2111032010006)

Assign

Functional | Sprint 4 | Collaborate | Epic 4

Bucket	Progress	Priority
Product Backlog	Completed	Important

Start date	Due date	Repeat
09/09/2024	24/09/2024	Does not repeat

Notes

Show on card

User Story 4: Implement Real-Time DDoS Mitigation in NS3

- **Title:** Implement Real-Time DDoS Mitigation in NS3
- **Description:** As a security engineer, I want to integrate the GNN model with the NS3 simulation to enable real-time detection and mitigation of DDoS attacks, ensuring the network remains secure and operational.
- **Points:** 10 points (complexity: very high, effort: extensive)
- **Target Audience:**
  - Security engineers working on real-time threat detection and mitigation.
  - Network administrators maintaining the security of IoT networks.
  - Researchers focused on dynamic security solutions for IoT environments.
- **Estimation of Effort:**
  - **Integration of Model with NS3:** 4 days
  - **Implementation of Packet Filtering Mechanism:** 4 days
  - **Testing and Validation of Mitigation Strategy:** 7 days
  - **Documentation:** 1 day
  - **Total Effort:** 16 days

Acceptance Criteria:

- The GNN model is successfully integrated with the NS3 simulation environment.
- The simulation detects malicious traffic in real-time and mitigates the DDoS attack by dropping packets from identified malicious IPs.
- The network performance metrics (e.g., throughput, latency) are monitored, showing that legitimate traffic remains unaffected.
- The real-time mitigation strategy is documented, including results from various test scenarios.

Checklist 6 / 6

Show on card

3. Functional Documents

a. User Story 1

- **Introduction**

The objective of this user story is to set up the NS-3 simulation environment necessary for the AI-driven dynamic fuzz testing framework, which will be used to generate realistic network traffic, including benign and DDoS attack data. This step is crucial for training and validating the Graph Neural Network (GNN) model in subsequent tasks.

- **2. Product Goal**

The goal is to establish a fully operational NS-3 simulation environment to simulate IoT network scenarios. The environment will be used to create datasets that include both benign and DDoS attack traffic, which are essential for training the AI model.

- **Demography (Users Location)**
    - **Target Users:** Developers and researchers working on IoT security.
    - **User Characteristics:** Individuals with technical expertise in network simulations and AI-driven security frameworks.
    - **Location:** Global usage with an emphasis on research and academic environments.

- **Business Processes**
    - **Simulation Environment Setup:**
        - Install NS-3 on a Linux-based system.
        - Integrate necessary libraries and modules for IoT simulations.

        - Configure network topologies to mimic IoT environments.
    - **Data Generation:**
        - Execute network simulations to generate traffic data.
        - Capture traffic logs in a format suitable for further analysis and model training.

- **Features**
    - **NS-3 Installation and Configuration:**
        - Install NS-3 on the chosen platform.
        - Ensure compatibility with necessary network protocols and IoT configurations.
    - **Network Topology Setup:**
        - Design and implement network topologies that simulate IoT networks.
    - **Traffic Log Generation:**
        - Capture network traffic in logs for analysis and model training.

- **Authorization Matrix**

ROLE	**Access Level**
Developer	Full access to configure and run NS-3 simulations
Researcher	Access to network traffic logs and simulation results.
Admin	Full access to system and simulation environment.

- **Assumptions**

  - The development environment remains stable during the setup process.
  - The team has access to necessary hardware resources for running simulations.
  - Necessary libraries and dependencies are available and compatible with NS-3.

- **Target Audience**
  - **Audience:** Developers, Researchers, Academic Institutions.
  - **Effort Estimation:** Approximately 3 days to 1 week, depending on complexity and resource availability.

- **Acceptance Criteria**
  - NS-3 is successfully installed and configured on the system.
  - Network topologies representing IoT networks are implemented.
  - Simulation runs successfully, generating traffic logs in the desired format.

- **Checklist**
  - NS-3 installed and configured.
  - Necessary libraries and dependencies integrated.
  - Network topologies designed and implemented.
  - Traffic logs generated and verified.

b. **User Story 2**

- **Introduction**

This user story focuses on generating the dataset needed for training the GNN model. The dataset will include traffic data from IoT simulations, comprising both benign and DDoS attack traffic. This data is crucial for developing a robust AI-driven security framework capable of detecting and mitigating DDoS attacks in real-time.

- **Product Goal**

The goal is to generate a labelled dataset from the NS-3 simulation environment, which will be used to train the GNN model for identifying and mitigating DDoS attacks in IoT networks.

- **Demography (Users Location):**

    a. Target Users: AI developers, security analysts, researchers.

    b. User Characteristics: Individuals with experience in AI model training and network security.

    c. Location: Academic and research institutions globally.

- **Business Processes:**

    - **Data Collection:**

        a. Run network simulations to generate diverse traffic patterns.

        b. Capture network logs in XML format.

    - **Data Processing:**

        a. Convert XML logs to CSV format.

        b. Label data entries as either benign or DDoS traffic.

- **Data Balancing:**
    a. Ensure the dataset has a balanced representation of benign and DDoS traffic.

- **Features:**
    - Traffic Simulation: Execute simulations in NS-3 to generate IoT traffic data.
    - Data Conversion: Convert and process XML logs to CSV for model training.
    - Data Labelling: Label data entries based on the traffic source as benign or DDoS.

- **Authorization Matrix:**

Role	Access Level
Developer	Access to raw simulation data and conversion tools.
Researcher	Access to the labelled dataset for analysis.
Data Scientist	Full access to processed and labelled datasets.

- **Assumptions:**
    - NS-3 simulations are run successfully without errors.
    - The XML to CSV conversion script is functional and accurate.

- Labelling criteria are clearly defined and adhered to during data processing.

- **Target Audience:**
  - **Audience**: Data Scientists, AI Developers, Researchers.
  - **Effort Estimation**: Approximately 1 week for complete dataset generation and processing.

- **Acceptance Criteria:**
  - Simulation data is captured and logged correctly.
  - XML logs are converted to CSV format without errors.
  - Data is accurately labelled as benign or DDoS traffic.
  - The dataset is balanced and ready for model training.

- **Checklist:**
  - Traffic data generated via NS-3 simulations.
  - XML logs converted to CSV format.
  - Data entries labelled correctly.
  - Dataset reviewed for balance and accuracy.

c. User Story 3

- **Introduction**

This document outlines the functionality required for training a Graph Neural Network (GNN) model for detecting Distributed Denial-of-Service (DDoS) attacks in IoT network traffic. This is part of a larger AI-driven IoT security project focusing on dynamic fuzz testing and mitigation of cyberattacks.

- **Product Goal**

The primary goal is to accurately detect and classify DDoS traffic using a GNN model trained on pre-processed datasets generated from IoT network simulations. The model will differentiate between benign and malicious traffic, aiding cybersecurity efforts in real-time threat detection.

- **Demography (Users and Locations)**
  - Target Users Data scientists, cybersecurity analysts, and AI researchers.
  - User Characteristics Proficient in network security, AI modelling, and working knowledge of GNNs.
  - Location Intended for global use by professionals and researchers involved in cybersecurity.

- **Business Processes**
  - Model Architecture Design
    - Define the architecture for the GNN model, considering input features like network traffic flow and topology.
    - Implement layers tailored for anomaly detection.

- Model Training
  - Use the pre-processed dataset to train the GNN model to recognize DDoS traffic.
  - Split the dataset into training, validation, and test sets.
- Model Testing and Saving
  - Evaluate the model on a separate validation dataset to assess its detection accuracy.
  - Store the trained model for deployment and further testing.

- **Features**
  - Model Training and Evaluation
  - Training process using supervised learning on labelled IoT network traffic data.
  - Validation to ensure the model achieves at least 75% accuracy in detecting DDoS attacks.
  - Utilize cross-validation to ensure robustness.

- **Hyperparameter Optimization**
  - Various settings tested for optimal model performance such as:
    - Learning Rate
    - Epochs
    - Hidden Layers
    - Batch Size
    - Optimizer
    - Weight Initialization
    - Regularization
    - Activation Functions
    - Loss Function
    - Early Stopping
    - Number of Layers

- **Model Saving**
  - Save the final model for use in deployment environments, enabling real-time detection.

- **Authorization Matrix**

Role	Access Level
Data Scientist	Full access to model training and tuning processes
Analyst	Access to trained model and its outputs for threat analysis
Admin	Full access to system resources and document

- **Assumptions**
  - The dataset is pre-processed and contains relevant traffic patterns for benign and DDoS scenarios.
  - Adequate computational resources are available for training the GNN.
  - Model evaluation metrics (accuracy, precision, recall) are pre-defined for validation.

- **Target Audience**

Audience Data Scientists, AI Researchers, Cybersecurity Analysts.

- **Effort Estimation**
  - Model Architecture Design: 3 days
  - Model Training: 5 days
  - Hyperparameter Tuning: 7 days
  - Documentation: 1 day
  - Total: 16 days

- **Acceptance Criteria**
  - The GNN model achieves at least 75% accuracy in detecting DDoS attacks.
  - Hyperparameters are tuned to optimize performance.
  - The model differentiates between DDoS and benign traffic.
  - Training procedures are well documented, and the trained model is saved for deployment.

- **Checklist**
  - Model architecture designed and implemented.
  - Dataset pre-processed and ready for training.
  - GNN model trained and validated.
  - Hyperparameters tuned to optimize detection performance.
  - Model saved for deployment.
  - Documentation completed.

d. User Story 4

- **Introduction:**

This user story focuses on integrating a Graph Neural Network (GNN) model with the NS-3 simulation environment to enable real-time detection and mitigation of DDoS attacks. The objective is to ensure network security and operational efficiency while mitigating malicious traffic through dynamic packet filtering.

- **Product Goal:**

The aim is to enhance the NS-3 environment by integrating AI-powered DDoS detection, utilizing the GNN model to analyse traffic patterns and implement a real-time mitigation strategy that drops malicious traffic while maintaining legitimate network flow.

- **Demography (Target Audience):**
    - **Security Engineers:** Focused on real-time threat detection and mitigation.
    - **Network Administrators:** Responsible for securing IoT networks.
    - **Researchers:** Investigating dynamic security solutions in IoT environments.

- **Business Processes:**
    - **GNN Model Integration with NS-3:**
        a. Modify the existing NS-3 environment to interface with the GNN model.
        b. Train the GNN model using IoT traffic data to detect anomalies such as DDoS attacks.
        c. Implement real-time traffic analysis through the model to monitor for malicious activity.

- **DDoS Mitigation Strategy:**
    a. Employ dynamic packet filtering to block malicious IP addresses.
    b. Ensure minimal impact on legitimate traffic by optimizing the filtering mechanism.
    c. Monitor network metrics (latency, throughput) during mitigation.

- **Features:**
    - **Integration of GNN with NS-3:**
        - Establish communication between NS-3 and the trained GNN model.
        - Facilitate real-time traffic analysis during simulation runs.
    - **Dynamic Packet Filtering:**
        - Real-time detection and packet drop for malicious IP addresses.
        - Monitor and optimize the performance of the packet-filtering mechanism.
    - **Monitoring Network Performance:**
        - Measure throughput, latency, and other performance indicators to ensure network stability.

- **Roles & Authorization Matrix:**

Role	Access Level
Security Engineer	Full access to configure real-time detection and mitigation.
Network Administrator	Monitoring access to ensure network security is maintained.
Researcher	Access to performance data and logs for testing purposes.

- **Assumptions:**
    - The NS-3 simulation environment is stable and configured for IoT simulations.
    - The GNN model has been trained with relevant datasets (including benign and DDoS traffic).
    - Packet filtering libraries and dependencies are compatible with NS-3.

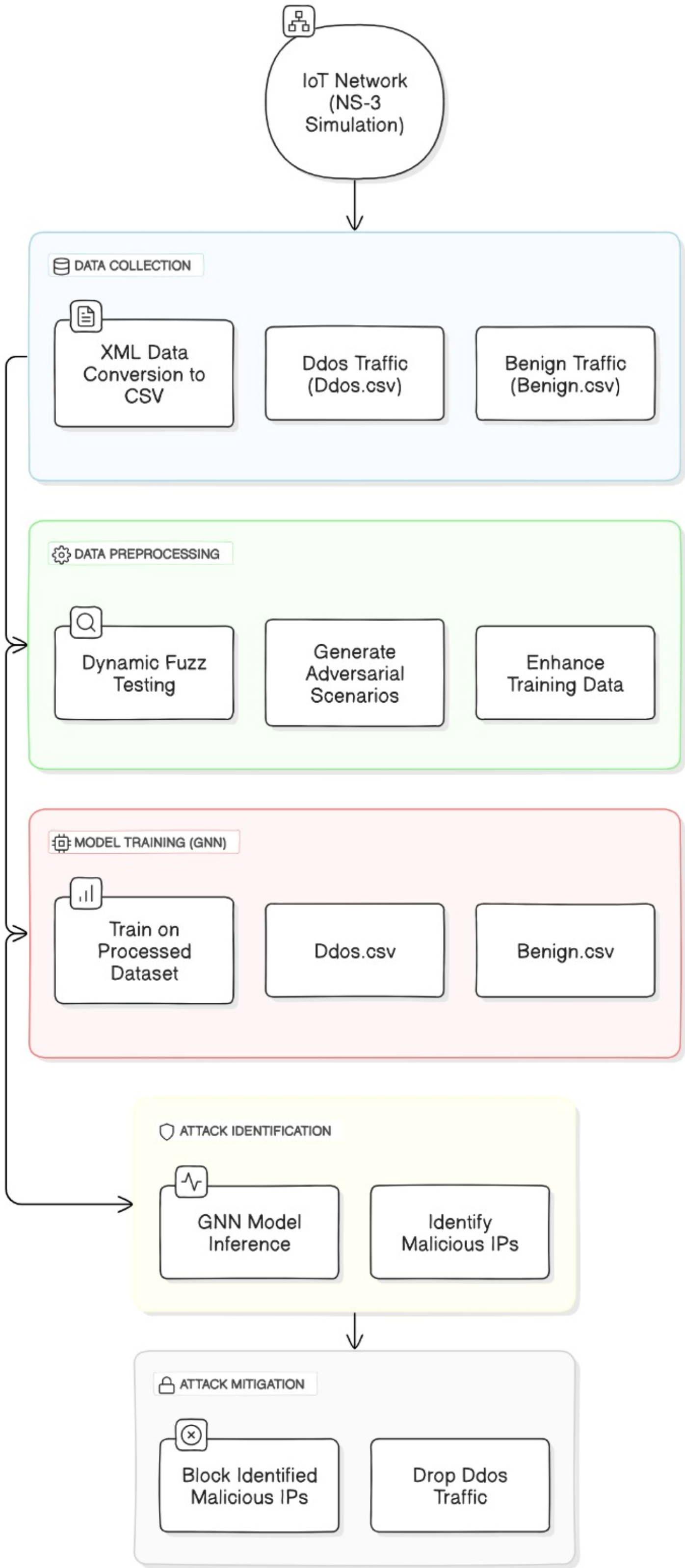
- **Effort Estimation:**
    - **GNN Model Integration with NS-3:** 4 days
    - **Packet Filtering Mechanism Implementation:** 4 days
    - **Testing & Validation:** 7 days
    - **Documentation:** 1 day
    - **Total:** 16 days

- **Acceptance Criteria:**
  - Successful integration of the GNN model with NS-3.
  - Real-time detection of DDoS attacks and mitigation through packet filtering.
  - The network performance remains stable with legitimate traffic unaffected.
  - Full documentation of the real-time mitigation strategy and test results.

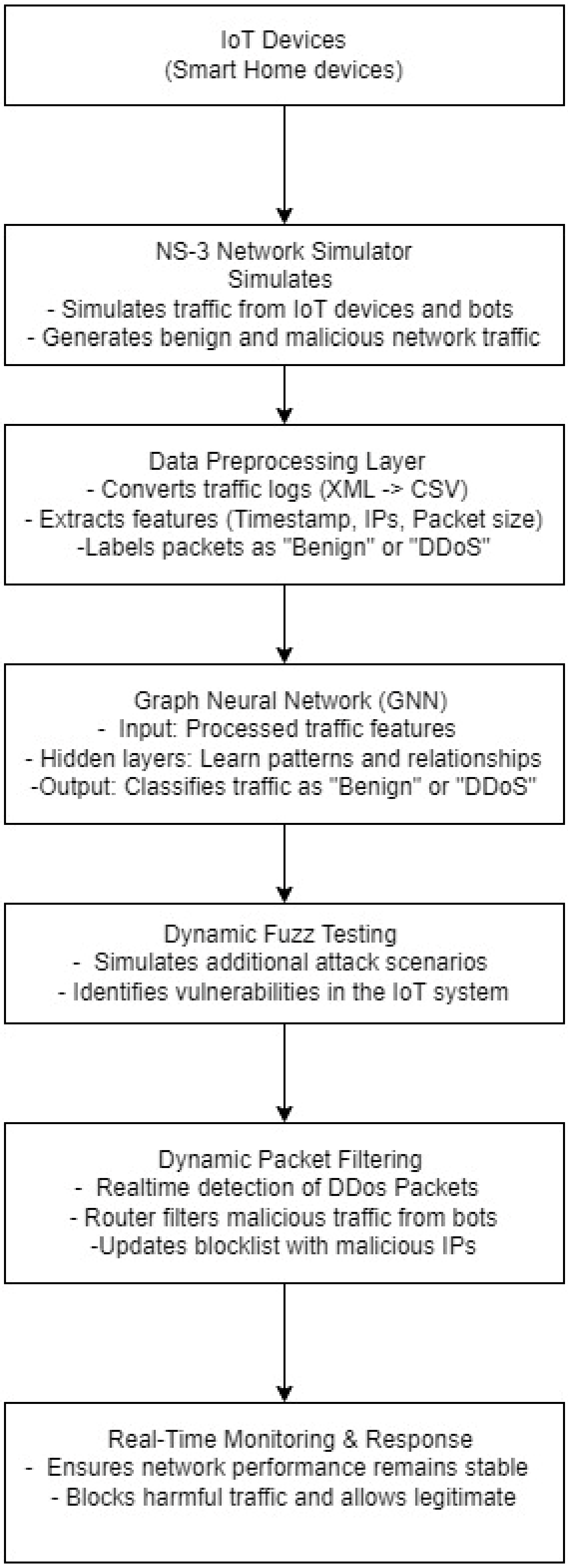
- **Checklist:**
  - GNN model integrated with NS-3.
  - Real-time DDoS detection implemented.
  - Packet filtering for malicious traffic configured and operational.
  - Network performance metrics logged and analysed.
  - Documentation of results and findings completed.

4. Architecture Document

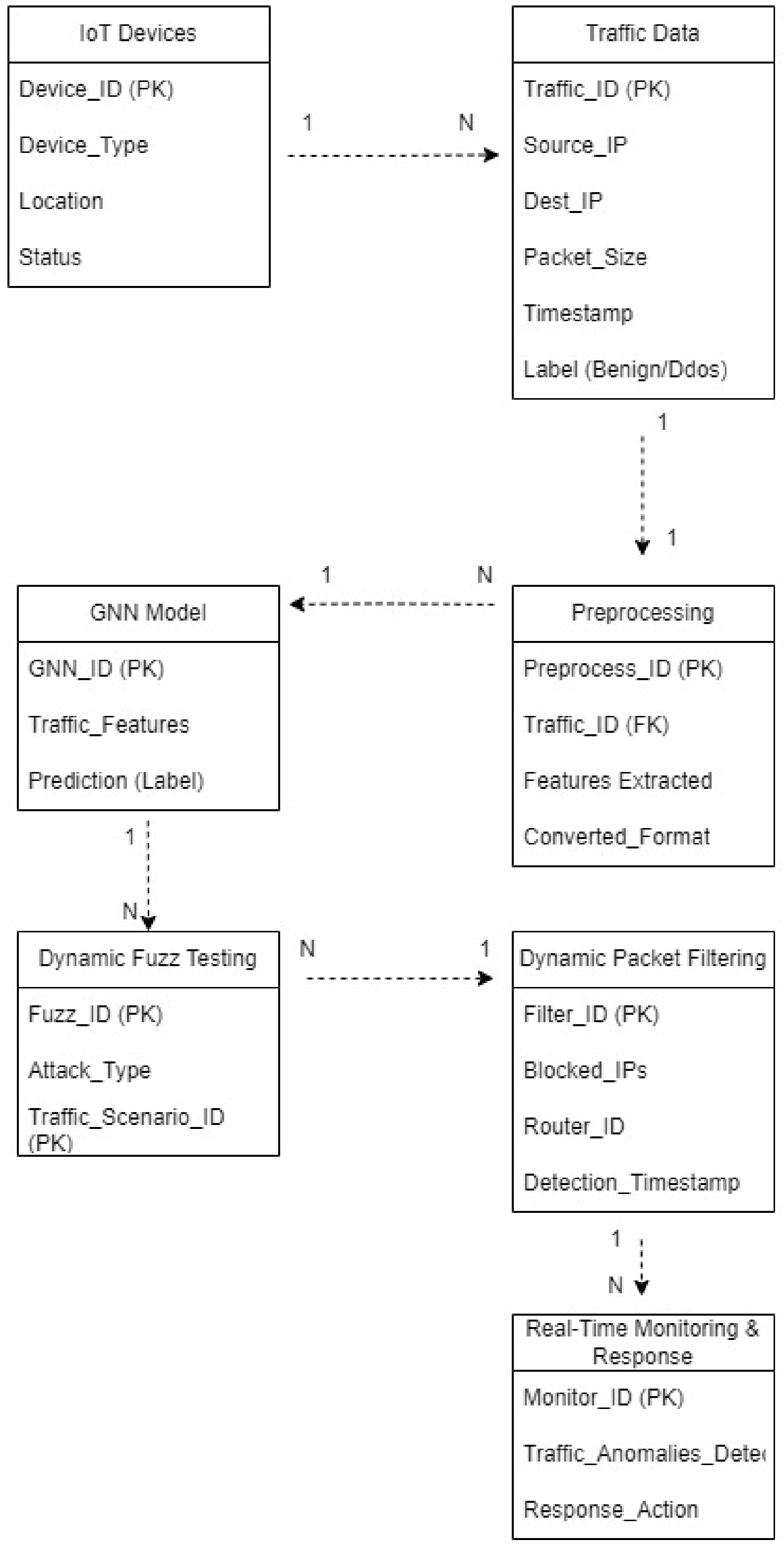
- **Architecture Diagram:**



- **Scheme Diagram:**



- **E-R Diagram:**


IoT Devices
Device_ID (PK)
Device_Type
Location
Status
1
N
Traffic Data
Traffic_ID (PK)
Source_IP
Dest_IP
Packet_Size
Timestamp
Label (Benign/Ddos)
1
1
1
N
GNN Model
GNN_ID (PK)
Traffic_Features
Prediction (Label)
Preprocessing
Preprocess_ID (PK)
Traffic_ID (FK)
Features Extracted
Converted_Format
1
N
Dynamic Fuzz Testing
Fuzz_ID (PK)
Attack_Type
Traffic_Scenario_ID (PK)
N
1
Dynamic Packet Filtering
Filter_ID (PK)
Blocked_IPs
Router_ID
Detection_Timestamp
1
N
Real-Time Monitoring & Response
Monitor_ID (PK)
Response_Action

5. Sprint Retrospective Document

	Sprint 1 : Setup NS3 Simulation Environment		
Liked	**Learned**	**Lacked**	**Longed For**
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
Successfully set up the NS3 environment without any major errors.	Learned the nuances of configuring an IoT network in NS3.	Lacked sufficient examples for simulating complex IoT devices.	Desired more efficient methods for configuring and simulating diverse IoT environments.
Collaboration between network engineers and data scientists led to efficient environment configuration.	Gained insights into how network simulations can generate valuable data for GNN model training.	Missing support for advanced network configurations out-of-the-box in NS3.	Wished for a more integrated system for logging and analyzing simulation data.
The initial IoT network topology was established smoothly.	Discovered best practices for organizing simulation files and settings.	Faced delays due to a lack of clear guidelines for generating XML traffic logs.	Hoped for additional modules in NS3 for simulating real-time network behavior.
Realistic traffic generation was accurate as per simulation requirements.	Enhanced knowledge on the limitations of default NS3 modules and the need for customization.	Insufficient time was allocated for testing the simulation setup under various scenarios.	Longed for more structured sprint planning and resource allocation to avoid last-minute rushes.
Detailed documentation was well-structured and comprehensive.	Understood the importance of proper testing procedures in simulation environments.	More powerful computing resources would have made simulation faster.	Desired quicker feedback from the testing phase, as it took longer than anticipated.

	Sprint 2 : Generate Dataset for GNN Training		
Liked	**Learned**	**Lacked**	**Longed For**
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
The dataset was successfully extracted and processed for model training.	Gained experience in transforming raw simulation data into useful training datasets.	Lack of real-world IoT traffic patterns limited the diversity of the dataset.	Wanted more detailed simulation logs with additional network parameters.
Collaboration between network engineers and data scientists improved data quality.	Understood the importance of balancing datasets to avoid model bias.	Insufficient feature documentation slowed down the feature engineering process.	Desired automated tools to expedite data extraction and cleaning processes.
Feature engineering helped in deriving relevant insights from the simulation logs.	Learned how to extract relevant features (e.g., timestamps, packet sizes) for GNN training.	Limited access to automated tools for dataset balancing.	Hoped for easier integration of external data sources for richer training datasets.
Preprocessing steps ensured that the dataset was well-balanced and usable.	Realized the necessity of ensuring dataset consistency for effective model performance.	Lacked predefined templates for preprocessing and feature extraction.	Wished for more comprehensive test datasets to check the feature quality.
Documentation of the data extraction and preprocessing steps was clear and concise.	Explored techniques for handling missing data in network traffic logs.	More team communication was needed during the dataset validation process.	Desired quicker feedback cycles from the data validation phase to avoid bottlenecks.

	Sprint 3 : Train GNN Model for DDoS Detection		
Liked	**Learned**	**Lacked**	**Longed For**
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
The model training process was smooth, and early results were promising.	Understood the impact of hyperparameters on GNN performance.	Lacked real-time evaluation during the model testing phase.	Wanted faster results from hyperparameter tuning using better computational resources.
Team collaboration improved during hyperparameter tuning efforts.	Learned techniques to fine-tune the model for different network traffic patterns.	Limited computational power made the hyperparameter tuning slow.	Desired real-time traffic to test the model on live data.
Reached the target accuracy of 75% on the validation set.	Gained experience in handling large datasets during model training.	Faced challenges in finding optimal learning rates and other parameters.	Hoped for a more intuitive visualization of model performance over time.
The GNN model was able to classify DDoS traffic effectively.	Explored how GNN architectures can be customized for IoT traffic analysis.	Needed more test data with various DDoS attack patterns for robust training.	Wished for better tools to automate model performance monitoring.
Good progress was made in documenting model architecture and training procedures.	Realized the importance of validation in reducing overfitting during training.	Lack of comprehensive documentation on hyperparameter tuning strategies.	Desired quicker model validation feedback to avoid prolonged tuning cycles.

	Sprint 4 : Implement Real-Time DDoS Mitigation in NS3		
Liked	**Learned**	**Lacked**	**Longed For**
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
Successfully integrated the GNN model with NS3 for real-time mitigation.	Learned how to integrate a GNN model within a network simulation environment.	Lacked real-time logging tools to monitor the mitigation process more effectively.	Desired quicker ways to simulate different types of DDoS attacks in NS3.
The packet filtering mechanism worked as expected to block malicious traffic.	Gained insights into real-time packet filtering and its effects on network performance.	Required more comprehensive test cases to validate the mitigation strategy under varied conditions.	Wished for more advanced visualization tools to monitor traffic in real-time.
Team communication was efficient during the integration and testing phases.	Understood the importance of balancing security measures with network throughput.	Lacked sufficient documentation on integrating machine learning models in NS3.	Hoped for seamless integration of the mitigation strategy into live network environments.
Network performance was monitored closely, and legitimate traffic was unaffected.	Learned how to implement dynamic filtering based on the model's predictions.	Faced delays due to insufficient knowledge about real-time packet filtering techniques.	Wanted more advanced packet filtering options that are easily configurable.
Clear documentation of the mitigation strategy helped in replicating the process.	Realized the challenges of maintaining performance while mitigating attacks.	Required more scenarios to fully test the GNN model's effectiveness in diverse traffic conditions.	Desired more real-world IoT traffic data for more accurate mitigation testing.