**Project Design Phase-II**
**Technology Stack (Architecture & Stack)**

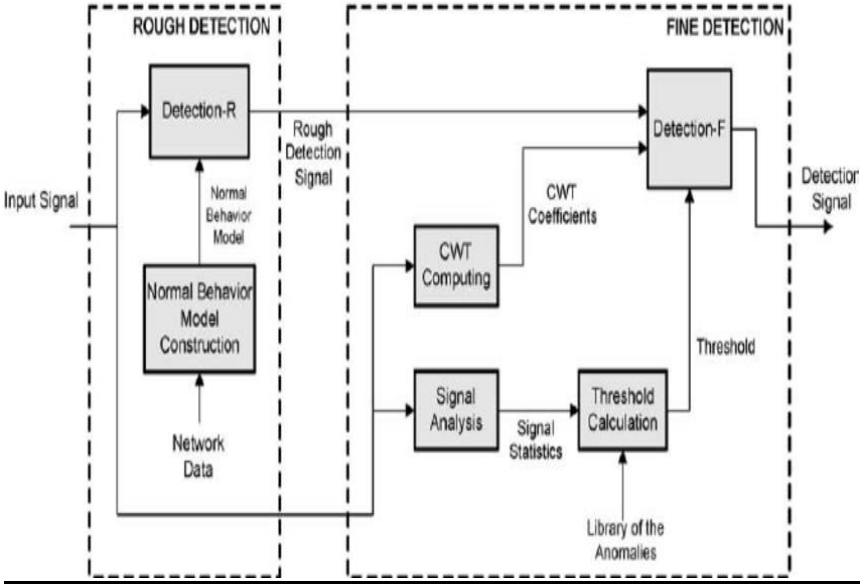
| | |
|---|---|
| Date | 19 September 2022 |
| Team ID | Team 4.2 |
| Project Name | Network anomaly detection |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Network Anomaly Detection**

**Reference:** https://www.researchgate.net/figure/Anomaly-detection-system-proposed-architecture_fig1_220065410




Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

**Table-1 : Components & Technologies:**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | Data Collection | Gathering network data for analysis | SNMP |
| 2. | Data Preprocessing | Cleaning and preparing data for analysis | Python Pandas |
| 3. | Feature Extraction | Extracting relevant features from network data | Scapy |
| 4. | Anomaly Detection Model | Building a model to detect anomalies in the network | LSTM (Long Short-Term Memory) |
| 5. | Alert Generation | Generating alerts for detected anomalies | Python SMTP |
| 6. | Data Visualization | Creating visual representations of network data | Matplotlib |
| 7. | Dashboard | Displaying real-time network monitoring and analysis | Grafana |
| 8. | Data Storage | Storing network data for future reference | Elasticsearch |
| 9. | Data Querying | Querying and retrieving specific network data | Kibana |
| 10. | Real-Time Monitoring | Monitoring network traffic and activity in real time | Kafka |
| 11. | Reporting | Creating comprehensive reports on network anomalies | Jupyter Notebooks |

**Table-2: Application Characteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | User Interface | Provides an interactive interface for users | React.js |
| 2. | Backend Server | Manages application logic and data processing | Node.js |
| 3. | Database Management | Stores and manages application data | MongoDB |

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 4. | Authentication | Ensures secure access for authorized users | JSON Web Tokens (JWT) |
| 5. | Scalability | Supports the ability to handle increased loads | Docker, Kubernetes |

**References:**


**https://chat.openai.com/**

**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**