TOPIC: IT OPERATIONS & NETWORK MANAGEMENT

By: A.G.SHERYL -21z348
R.SRINITHI - 21z358

Platform Architecture:

Data Collection Layer:

Sensors and Agents: Deployed across the network to capture real-time data such as traffic flow, packet details, and system logs.
Data Aggregator: Gathers data from various sensors and agents for processing.

Data Processing Layer:

Data Ingestion: The collected data is ingested into the system using tools like Apache Kafka or Flume for real-time data streams.
Data Storage: Data is stored in scalable databases such as Hadoop HDFS, Apache Cassandra, or cloud storage solutions for batch processing and historical analysis.
Preprocessing: Includes cleaning, normalizing, and transforming raw data. Tools like Apache Spark or Python's Pandas are used.

Analysis Layer:

Feature Engineering: Extracts relevant features from the raw data to enhance model performance.
Machine Learning Models: Deployed to detect anomalies and predict network failures. Models are trained using frameworks like TensorFlow, PyTorch, or Scikit-learn.

Inference Layer:

Real-time Analysis: Uses pre-trained models to analyze incoming data streams and detect anomalies in real-time.
Batch Analysis: Periodically processes historical data to refine models and detect patterns.

Visualisation and Reporting Layer:

Dashboards: Real-time monitoring dashboards built using tools like Grafana, Kibana, or custom web applications.
Alerts and Notifications: Integrates with messaging systems (e.g., Slack, email, SMS) to alert administrators of detected anomalies or failures.

User Interface Layer:

Web Interface: An interactive interface built using web technologies like React, Angular, or Vue.js, allowing users to monitor network status, view historical data, and receive alerts.
Mobile App: A companion mobile app for on-the-go monitoring and alerts.

Analysis Algorithms

Anomaly Detection:

Statistical Methods: Z-score, moving average, and other statistical techniques to identify deviations from normal patterns.
Machine Learning Algorithms:
Supervised Learning: Algorithms like Random Forest, SVM, and Neural Networks trained on labeled data to classify network traffic and detect failures.
Unsupervised Learning: Algorithms like K-means clustering, DBSCAN, and Isolation Forest to detect outliers and unknown anomalies in unlabeled data.
Deep Learning: LSTM networks for sequence prediction, autoencoders for anomaly detection.

Predictive Analytics:

Regression Analysis: Predicts future network load and potential failures based on historical data.
Time Series Analysis: ARIMA, Prophet, and other time series models to forecast network traffic trends and detect anomalies.

User Interface Design

Dashboard:

Overview Panel: Displays key metrics such as current network load, detected anomalies, and system health status.
Real-time Traffic Visualization: Graphs and charts showing live data on network traffic, packet loss, latency, and other critical metrics.
Alerts Panel: Lists recent alerts and notifications, allowing users to quickly see and respond to issues.

Data Visualization:

Historical Data Analysis: Interactive charts and graphs for exploring historical network data, filtering by time range, and drilling down into specific events.
Anomaly Heatmaps: Visual representations of where anomalies are occurring most frequently within the network.

Alerts and Notifications:

Configurable Alerts: Users can set thresholds and rules for generating alerts, choosing how and when to be notified.
Alert Details: Provides detailed information about each alert, including time, affected components, and suggested actions.

Configuration and Settings:

User Management: Admin interface for managing user roles, permissions, and access controls.
System Settings: Interface for configuring network monitoring parameters, data retention policies, and integration with other systems.

Conclusion

An AI-driven network management system encompasses a robust architecture, effective analysis algorithms, and a user-friendly interface. By leveraging machine learning models, real-time data processing, and intuitive visualizations, such a system can significantly enhance network monitoring capabilities and proactively detect and address network failures.