RAJALAKSHMI
ENGINEERING
COLLEGE

Department of Artificial Intelligence and Data Science

MediSpark: Optimizing Hospital Resource Utilization using PySpark on Databricks

Mr.Suresh Kumar
ASST.PROFESSOR

Kaviya P - 231801082

Problem Statement and Motivation

Traditional traffic management systems are largely **reactive**, responding to congestion after it has already formed. While IoT sensors provide a wealth of data, its sheer **Volume, Velocity, and Variety (3Vs)** overwhelm conventional databases and analytics tools. The core problem is the **inability of existing systems to perform real-time processing and predictive analytics on massive, continuous streams of sensor data**.

This project addresses the specific challenge of designing and implementing an **end-to-end Big**

MOTIVATION :

The motivation for this project stems from the severe and escalating negative impacts of traffic congestion:

Economic Impact: Wasted fuel and lost productivity hours cost national economies billions of dollars annually.

Environmental Impact: Idling and slow-moving vehicles are a major source of CO2 and other greenhouse gas emissions, contributing to air pollution and climate change.

Objectives

- To **design** a scalable Big Data architecture (e.g., Kappa Architecture) suita streams.
- To **implement** a data ingestion pipeline using Apache Kafka to collect and
- To **develop** a data processing layer using Apache Spark (Spark Streaming) aggregate the data.

Abstract

Urban traffic congestion is a critical global issue, causing significant economic losses, environmental pollution, and reduced quality of life. The proliferation of Internet of Things (IoT) sensors—such as GPS units, loop detectors, traffic cameras, and mobile devices—generates a massive, high-velocity, and diverse (Big Data) stream of real-time traffic information. However, traditional data processing systems cannot handle this scale. This project proposes a **scalable Big Data architecture** (based on the Kappa/Lambda model) to effectively **ingest, store, process, and analyze** this IoT data.

The system will provide accurate, short-term predictions, enabling proactive traffic management, dynamic route guidance, and smarter urban planning.

INTRODUCTION (SRS)

Urbanization is accelerating globally, leading to an unprecedented number of vehicles on city roads. This growth has far outpaced the development of new infrastructure, resulting in chronic traffic congestion. This congestion is not merely an inconvenience; it is a systemic problem with profound economic, environmental, and social costs.

Simultaneously, the rise of the **Internet of Things (IoT)** has instrumented our cities with a vast network of sensors. Connected vehicles, smartphones, road-side sensors, and cameras generate terabytes of data every day. This data contains the "digital footprints" of traffic flow and holds the key to understanding, modeling, and ultimately, predicting congestion.

However, this data is a classic **Big Data** challenge. Its high volume (terabytes), high velocity (streaming in real-time), and high variety (structured sensor readings, unstructured camera feeds) make it impossible to manage with traditional databases.

ALGORITHMS

LSTM (Long Short-Term Memory) Network.:

Traffic is a **time-series** problem. Today's traffic is highly dependent on yesterday's traffic (e.g., weekly patterns) and the last few hours of traffic (e.g., the buildup of rush hour). LSTMs are a type of RNN specifically designed to learn these long-term temporal dependencies, which simple models miss.

SYSTEM REQUIREMENTS

- **Operating System:** Linux (Ubuntu/CentOS recommended for server cluster).
- **Hadoop Ecosystem:** Hadoop 3.x (for HDFS), Apache Spark 3.x (Core, Streaming, MLlib, SQL).
- **Ingestion:** Apache Kafka 2.x.
- **Database:** Apache Cassandra 4.x or Apache HBase 2.x.
- **Programming:** Python 3.8+ (with libraries: PySpark, Pandas, NumPy, Scikit-learn, TensorFlow/Keras) or Scala.

FUTURE SCOPE

- **Data Fusion:** Integrate more data sources for higher accuracy, such as **weather forecasts**, **public event schedules** (concerts, games), **road construction schedules**, and real-time social media (e.g., Twitter data for accident reports).
- **Prescriptive Analytics:** Move beyond *prediction* (what will happen) to *prescription* (what should we do?). Use the model to automatically suggest **dynamic traffic light timing adjustments** or optimal re-routing strategies.
- **Edge Computing:** To reduce latency and data volume, perform simple pre-processing (like filtering) on "edge" devices (e.g., gateways at the traffic intersections) before sending data to the central Kafka cluster.

Thank You