IEEE

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING
STUDENT BRANCH

UCSC

IntelliHack 5.0

TEAM HYPER TUNERS

TASK 01 – Part 2

System Design

intelli Hack 5.0

Artificial . But Intelligent

IntelliHack 5.0 - Task 1 Part 2: Advanced MLOps System Design

Introduction

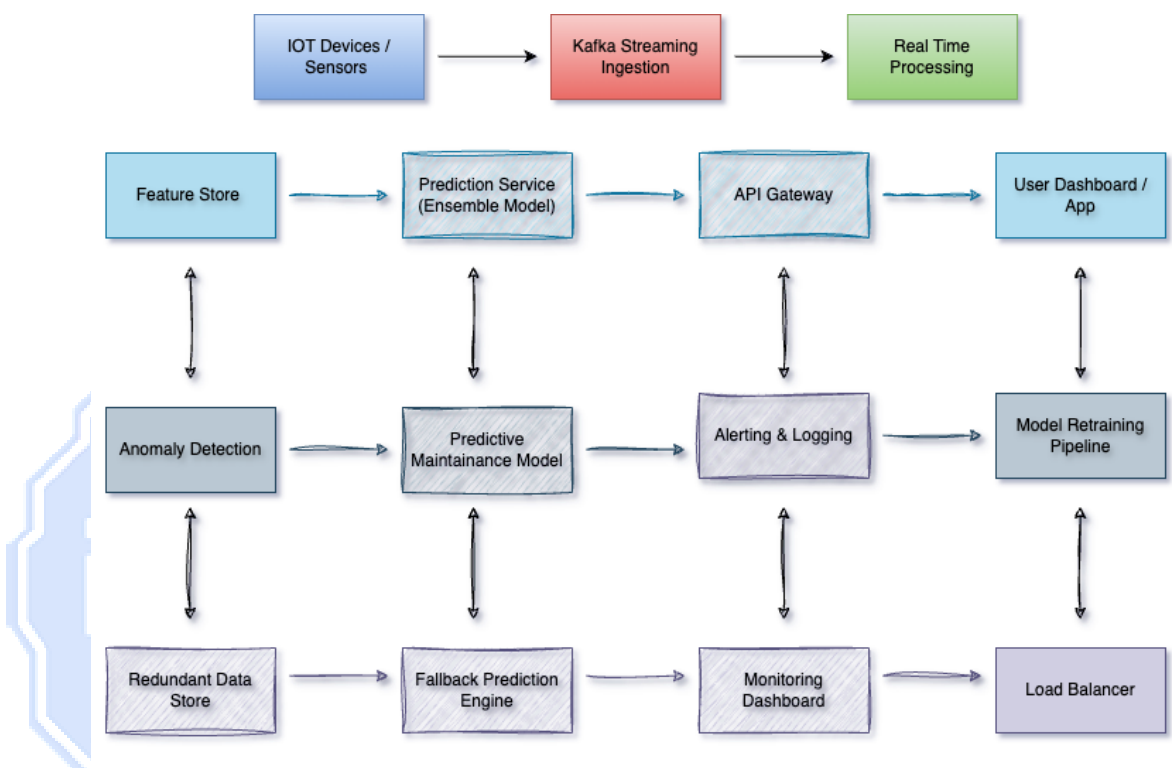
In the era of smart agriculture, accurate weather forecasting is critical for farmers to optimize irrigation, planting, and harvesting schedules. Traditional weather prediction methods often fall short in delivering hyper-local insights, leaving farmers vulnerable to unpredictable conditions. This challenge, presented by IntelliHack 5.0, tasks Team Hyper Tuners with developing an innovative solution to predict rain probability based on historical and real-time weather data, addressing the limitations of conventional forecasts.

For Part 2, we focus on designing an advanced MLOps system to predict the probability of rain for the next 21 days using real-time data from IoT sensors, collected at 1-minute intervals. The system must handle sensor malfunctions, ensure scalability, and deliver reliable forecasts to farmers. Leveraging the predictive model developed in Part 1, we integrate cutting-edge technologies such as Kafka streaming, predictive maintenance, and automated retraining. Our design emphasizes fault tolerance, real-time processing, and user accessibility, ensuring a production-ready solution for smart agriculture applications.

This report outlines our system architecture, highlighting innovative features to meet the competition's requirements. Team Hyper Tuners aims to deliver a transformative, scalable tool that empowers farmers with precise, localized weather predictions, positioning us as leaders in this challenge.

System Diagram

Below is the architecture for our MLOps system, illustrating the data flow from IoT devices to user delivery, with mechanisms for fault tolerance, scalability, and continuous improvement.



Component Descriptions

- **IoT Devices/Sensors**: Real-time weather sensors with self-diagnostic firmware.
- **Kafka Streaming Ingestion**: Scalable data ingestion with partitioned streams.
- **Real-Time Preprocessing**: Spark-based cleaning with dynamic outlier detection.
- **Feature Store**: Time-series feature repository for consistency.
- **Prediction Service**: Containerized ensemble model with A/B testing.
- **API Gateway**: Secure, cached endpoint for low-latency access.
- **User Dashboard/App**: Interactive forecast UI with push notifications.
- **Anomaly Detection**: Real-time malfunction flagging with Isolation Forest.
- **Predictive Maintenance Model**: LSTM-based failure prediction.
- **Alerting & Logging**: Prometheus-powered incident tracking.
- **Model Retraining Pipeline**: Weekly retraining with hyperparameter tuning.
- **Redundant Data Store**: Geospatial-enabled backup database.
- **Fallback Prediction Engine**: Seamless failover with lightweight model.
- **Monitoring Dashboard**: Real-time system health visualization.
- **Load Balancer**: Auto-scaling traffic distribution.

The diagram depicts a streamlined pipeline starting from IoT devices, moving through real-time data ingestion and preprocessing, feature storage, prediction, and delivery to users via an API gateway. It also includes advanced components for anomaly detection, predictive maintenance, automated retraining, and failover mechanisms to handle sensor malfunctions and ensure uninterrupted service.

Component Descriptions

- IoT Devices/Sensors: Real-time weather sensors with self-diagnostic firmware, collecting data like temperature and humidity every minute.

- Kafka Streaming Ingestion: Scalable data ingestion with partitioned streams, ensuring high throughput and fault tolerance.

- Real-Time Preprocessing: Spark-based cleaning with dynamic outlier detection, processing data within seconds.

- Feature Store: Time-series feature repository for consistency, supporting historical and real-time data.

- Prediction Service (Ensemble Model): Containerized ensemble model (LightGBM + Random Forest from Part 1) with A/B testing, deployed for low-latency predictions.

- API Gateway: Secure, cached endpoint for low-latency access, handling user requests efficiently.

- User Dashboard/App: Interactive forecast UI with push notifications, providing farmers with actionable insights.

- Anomaly Detection: Real-time malfunction flagging with Isolation Forest, detecting issues instantly.

- Predictive Maintenance Model: LSTM-based failure prediction, forecasting sensor issues to minimize downtime.

- Alerting & Logging: Prometheus-powered incident tracking, notifying engineers of malfunctions.

- Model Retraining Pipeline: Weekly retraining with hyperparameter tuning, adapting to new weather patterns.

- Redundant Data Store: Geo-spatial enabled backup database, ensuring data availability during outages.

- Fallback Prediction Engine: Seamless failover with lightweight model, maintaining service during sensor failures.

Conclusion

Our advanced MLOps system integrates real-time IoT data, robust fault tolerance, and automated retraining to deliver reliable 21-day rain probability forecasts. By incorporating predictive maintenance and scalable technologies like Kafka, we ensure resilience and adaptability, meeting the needs of smart agriculture. This design positions Team Hyper Tuners to deliver a transformative solution for farmers, with potential for broader applications in weather forecasting.

Acknowledgement

We, Team Hyper Tuners, extend our sincere gratitude to the UCSC team for organizing the IntelliHack 5.0 competition. Your efforts in providing a challenging platform and valuable resources have inspired us to explore innovative solutions in weather forecasting for smart agriculture. Thank you for this opportunity to grow and compete. 🔥

Team Members: -

- Janitha Rajapaksha
- Sanjula Weerasekara
- Hasindu Nimesh
- Veenave Samarasinghe