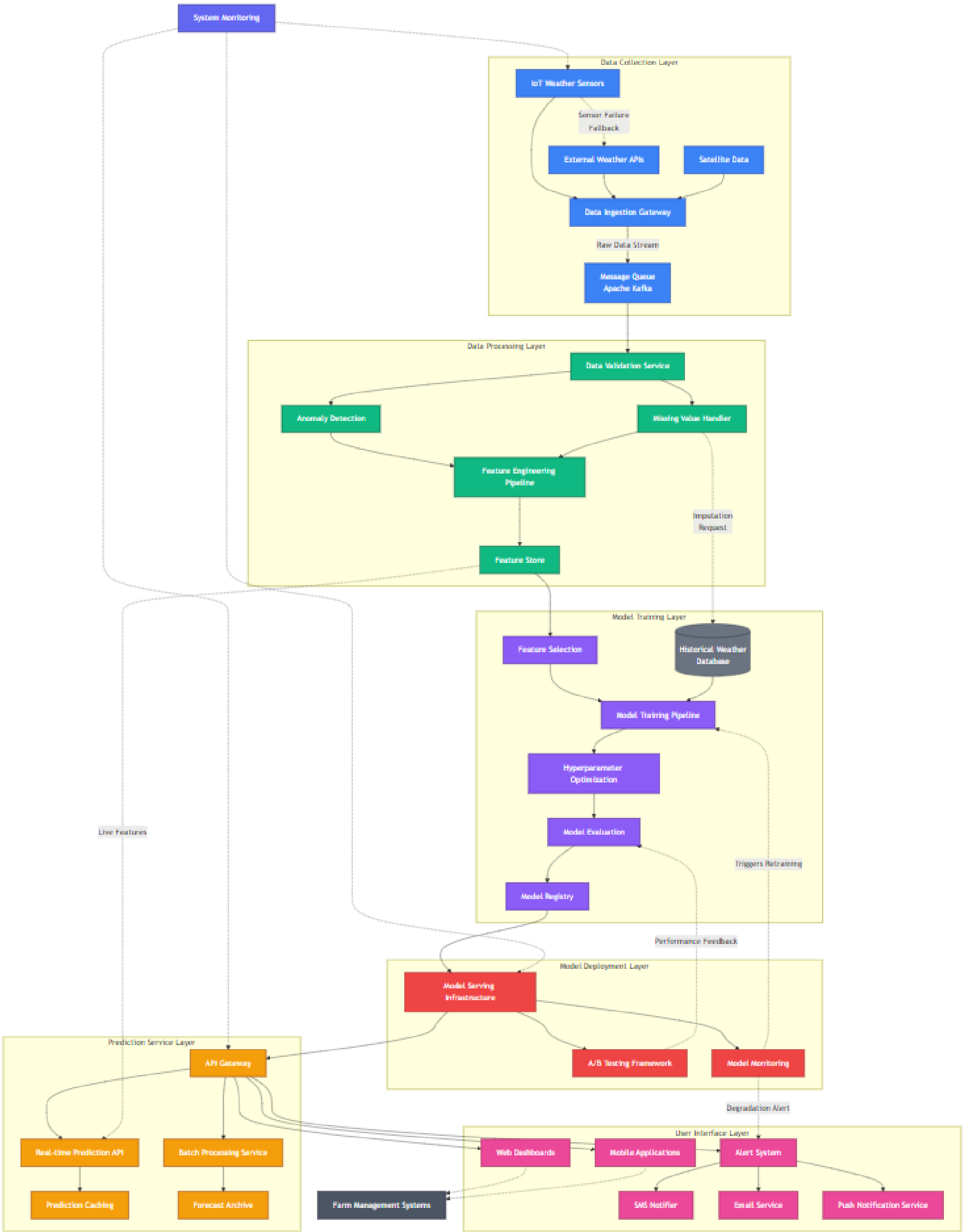
Part 2: MLOps System Architecture - Smart Agriculture Weather Forecasting System



1. Executive Summary

This report outlines a comprehensive MLOps architecture designed for hyper-local rainfall prediction in agricultural settings. The system addresses the challenge of providing reliable 21-day weather forecasts while handling sensor malfunctions and data quality issues. Key features include real-time processing of 1-minute interval sensor data, conditional imputation of missing values, ensemble machine learning models optimized for rainfall prediction, and intuitive user interfaces tailored for farming applications. The architecture employs sophisticated fault-tolerance mechanisms to ensure continuous operation even when sensors fail, making it particularly suitable for smart agriculture deployments in remote areas.

2. System Architecture

The architecture follows a layered design with six primary components working together to transform sensor readings into actionable agricultural insights:

Data Collection Layer

- **IoT Weather Sensors**: Distributed network collecting temperature, humidity, pressure, wind speed, and precipitation at 1-minute intervals
- **Data Ingestion Gateway**: Manages communication with sensors, implements buffering for network interruptions
- **External Weather APIs & Satellite Data**: Alternative data sources activated during sensor failures
- **Message Queue (Apache Kafka)**: Streams raw data to processing layer with fault-tolerant design

Data Processing Layer

- **Data Validation Service**: Applies physical constraints and statistical validation to detect anomalies
- **Missing Value Handler**: Implements conditional imputation based on spatial, temporal, and meteorological patterns
- **Feature Engineering Pipeline**: Generates derived features including rolling statistics, lag values, and trend indicators
- **Feature Store**: Centralizes feature management with versioning and lineage tracking

Model Training Layer

- **Model Training Pipeline**: Implements ensemble approaches with SMOTE resampling to address class imbalance
- **Hyperparameter Optimization**: Uses Bayesian methods to fine-tune model configurations
- **Model Evaluation**: Applies time-series cross-validation to simulate real-world forecasting
- **Model Registry**: Maintains comprehensive metadata on all trained models

Model Deployment Layer

- **Model Serving Infrastructure**: Containerized deployment ensuring consistency across environments
- **A/B Testing Framework**: Controlled comparison of model versions with automatic promotion
- **Model Monitoring**: Detects performance degradation and data drift, triggering retraining

Prediction Service Layer

- **API Gateway**: Provides secure access with authentication and rate limiting
- **Real-time Prediction API**: Delivers on-demand forecasts with configurable prediction horizons
- **Batch Processing Service**: Generates scheduled forecasts for integration with farm management systems
- **Forecast Archive**: Maintains historical predictions for accuracy assessment

User Interface Layer

- **Web Dashboards & Mobile Applications**: Visualization platforms with location-aware predictions
- **Alert System**: Configurable notifications via SMS, email, and push notifications
- **Farm Management Integration**: Connects weather insights to agricultural planning tools

3. Handling Sensor Malfunctions

The system employs multiple strategies to maintain reliable predictions despite sensor failures:

1. **Multi-level Redundancy**: Critical locations are equipped with redundant sensors measuring the same parameters, minimizing the risk of complete data loss.
2. **Automatic Fallback Paths**: When primary sensors fail, the system automatically activates alternative data sources including nearby sensors, regional weather APIs, and satellite imagery.
3. **Intelligent Imputation**: Missing values are handled through sophisticated conditional imputation that considers:
    - Spatial correlations with nearby functioning sensors
    - Temporal patterns including seasonal and diurnal cycles
    - Weather physics constraints maintaining relationships between parameters
4. **Confidence Communication**: When predictions rely on imputed or lower-quality data:
    - Forecast confidence levels are automatically adjusted
    - Visual indicators in user interfaces clearly identify affected predictions

    - Alert messages include appropriate caveats about reliability
5. **Graceful Degradation**: As data quality decreases, the system systematically falls back to:
    - Specialized models trained for partial data scenarios
    - Regional forecasts with wider geographical scope
    - Historical pattern-based predictions for extreme cases
6. **Continuous Monitoring**: Automated health checks:
    - Monitor sensor heartbeats to quickly detect failures
    - Assess data quality metrics in real-time
    - Generate maintenance alerts for field technicians
    - Track repair statuses and estimated restoration times

This multi-faceted approach ensures that farmers receive continuous forecasting support even under suboptimal conditions, with appropriate transparency about prediction confidence.

4. Implementation Details

Data Collection Implementation

- Sensors include local storage capable of buffering 7 days of measurements during connectivity outages
- Solar power with battery backup provides 72+ hours of autonomous operation
- Heartbeat signals transmitted every 5 minutes allow rapid failure detection
- Data transmission uses low-power protocols (LoRaWAN) for remote deployment

Data Processing Implementation

- Apache Kafka streams process 1-minute interval data with partitioning for horizontal scaling
- Anomaly detection combines statistical methods, physical constraints, and ML classification
- Feature store implements both batch computation for training and on-demand calculation for inference
- Time series alignment synchronizes data from different sources with varying collection frequencies

Machine Learning Implementation

- Primary model: Gradient Boosting with SMOTE resampling (F1: 0.82, Recall: 0.91)
- Model training containerized using Docker and orchestrated via Airflow
- Retraining triggers: 1) Weekly scheduled updates, 2) Performance degradation, 3) Significant data drift
- Hyperparameter optimization balances accuracy with inference speed requirements

Deployment Implementation

- Containerized microservices architecture deployed on Kubernetes
- Blue-green deployment strategy for zero-downtime updates
- Automated canary analysis evaluates new models before full rollout
- Circuit breakers prevent cascading failures during service disruptions

User Interface Implementation

- Progressive web applications providing consistent experience across devices
- Offline capability with local storage of recent forecasts
- Configurable alerts based on rainfall probability thresholds and agricultural activities
- Data visualization optimized for both high-end devices and basic feature phones

5. Conclusion

The MLOps architecture presented here delivers reliable rainfall predictions for agricultural planning despite the challenges of sensor malfunctions and weather variability. By implementing sophisticated data processing, machine learning, and resilience mechanisms, the system provides continuous support for critical farming decisions. The 21-day forecasts enable optimal scheduling of irrigation, planting, and harvesting activities, potentially increasing yields while reducing resource waste. Future enhancements will focus on incorporating additional data sources, extending the prediction horizon, and developing crop-specific recommendation systems.