IoT-Driven Predictive Analytics for Resilient Supply Chains

Hemlatha Kaur Saran, Tanvi Singh

MS - Applied Artificial Intelligence, University of San Diego

AAI 530 IN1: Data Analytics and Internet of Things

Haisav Choksi

23rd Feb 2026

Abstract

Agricultural supply chains are increasingly vulnerable to climate variability, operational disruptions, and inefficient resource allocation. This study presents an Internet of Things (IoT)-driven predictive analytics framework designed to enhance resilience in smart greenhouse environments. Using real-time sensor data including temperature, humidity, water level, and soil nutrients (N, P, K), the system integrates advanced time-series preprocessing, stacked Long Short-Term Memory (LSTM) networks, Seasonal AutoRegressive Integrated Moving Average (SARIMA) modeling, and Autoencoder-based anomaly detection.

The dataset was resampled at five-minute intervals and engineered with rolling statistics and temperature differencing. A 60-step sliding window was used to train a stacked LSTM model with EarlyStopping to prevent overfitting. Model performance was evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R^2 metrics. Results indicate that the LSTM model outperformed the SARIMA baseline in capturing nonlinear temporal dependencies. An Autoencoder model was further implemented to detect anomalies, enabling the computation of a resilience score that reflects operational stability.

The findings demonstrate that integrating IoT data with deep learning and statistical forecasting models significantly enhances predictive accuracy and supply chain resilience in precision agriculture systems.

Introduction

The U.S. agricultural sector, valued at over $1.5 trillion (Chen, 2025), faces increasing disruptions due to climate change, labor shortages, and global trade instability. These disruptions directly impact supply chain continuity, crop yield stability, and operational costs. To address these challenges, real-time monitoring through Internet of Things (IoT) technologies offers significant opportunities for proactive decision-making.

IoT-enabled greenhouses deploy environmental sensors to continuously monitor temperature, humidity, water levels, and soil nutrient content. However, raw sensor data alone does not guarantee improved resilience. The transformation of sensor streams into predictive insights requires advanced analytics capable of modeling temporal dependencies and detecting anomalies.

This project develops an IoT-driven predictive framework that combines:

- Time-series preprocessing and feature engineering
- Deep learning (stacked LSTM) forecasting
- SARIMA statistical baseline comparison
- Autoencoder-based anomaly detection
- Resilience score computation

The goal is to evaluate whether predictive analytics can enhance agricultural supply chain stability and reduce operational risk.

Literature Review

IoT in Agriculture and Logistics

IoT systems have transformed agricultural monitoring by enabling real-time tracking of environmental conditions (Beg et al., 2025). Soil moisture sensors, humidity detectors, and temperature gauges allow continuous environmental assessment. In logistics, IoT enhances supply chain transparency through real-time shipment monitoring and predictive maintenance.

Predictive Analytics in Supply Chains

Machine learning models have been widely applied in demand forecasting and disruption prediction (Michael & William, 2025). Classical approaches such as ARIMA and SARIMA rely on linear assumptions and seasonal decomposition but struggle with nonlinear patterns.

Deep Learning for Time-Series Prediction

Long Short-Term Memory (LSTM) networks effectively model long-term dependencies in sequential data (Hua et al., 2019). Unlike traditional statistical models, LSTMs capture nonlinear relationships, making them well-suited for IoT sensor streams.

IoT-Enabled Precision Agriculture

IoT-based greenhouse monitoring systems integrate environmental sensing with automated control (Liu et al., 2024). However, predictive analytics integration remains underexplored in resilience assessment.

IoT System Design

Sensors

The greenhouse system includes:

- Temperature sensors
- Humidity sensors
- Water level sensors
- Soil nutrient sensors (N, P, K)

These sensors collect continuous environmental data.

Data Transmission

Data is transmitted using lightweight IoT protocols such as MQTT or HTTP (Eldho K J, 2025). Cloud storage platforms enable scalable data management.

Data Processing Pipeline

The processing framework includes:

1. Time-based resampling (5-minute intervals)
2. Time interpolation for missing values
3. Feature engineering
4. Normalization using MinMaxScaler
5. Sliding window sequence generation
6. Model training and evaluation.

Data Analysis and Methodology

Data Cleaning

- Converted date column to datetime index
- Removed duplicates
- Resampled to 5-minute intervals
- Applied time-based interpolation

Feature Engineering

Engineered features include:

- Temperature difference (first derivative)
- Rolling mean (5-step window)
- Risk label based on threshold exceedance

These features improve model sensitivity to environmental shifts.

Sequence Preparation

A sliding window of 60 time steps was used to generate sequences for LSTM training. This captures temporal dependencies across one-hour intervals.

LSTM Model

A stacked LSTM architecture was implemented:

- LSTM (64 units, return_sequences=True)
- LSTM (32 units)
- Dense (16 units, ReLU)
- Output layer (1 neuron)

EarlyStopping (patience=5) was used to prevent overfitting.

SARIMA Model

A SARIMA (1,1,1) (1,1,1,12) model was trained as a statistical baseline.

Autoencoder for Anomaly Detection

A feedforward Autoencoder was trained to reconstruct input sequences.

Reconstruction error exceeding the 95th percentile threshold was classified as anomalous.

Evaluation Metrics

Models were evaluated using:

- RMSE
- MAE
- R^2

Results and Discussion

Correlation Analysis

Feature correlation analysis revealed strong dependencies between temperature and humidity, supporting multivariate modeling.

LSTM Performance

The LSTM model demonstrated stable convergence with decreasing training and validation loss. It captured nonlinear fluctuations in greenhouse temperature effectively.

Key metrics:

- RMSE: Lower than SARIMA
- MAE: Significantly reduced
- R^2: High explanatory power

SARIMA Performance

SARIMA captured seasonal patterns but struggled with abrupt nonlinear changes. Prediction lag was observed during rapid environmental fluctuations.

Model Comparison

Model	RMSE	MAE	R2
LSTM	0.579315	0.563225	0.947868
SARIMA	2.674799	2.388138	-0.109396

The LSTM model outperformed SARIMA in predictive accuracy.

Anomaly Detection

Autoencoder reconstruction error analysis identified operational anomalies corresponding to sudden environmental shifts. These anomalies signal potential supply chain disruptions.

Resilience Score

A resilience score was computed using:

- Risk label frequency
- Anomaly frequency

Higher resilience scores correspond to stable environmental conditions and lower disruption probability.

Conclusion and Future Work

This study demonstrates that integrating IoT sensor data with advanced predictive analytics significantly enhances agricultural supply chain resilience. The stacked LSTM model outperformed the SARIMA statistical baseline in forecasting environmental variables. The addition of Autoencoder-based anomaly detection further improved operational awareness.

The computed resilience score provides a quantitative measure of system stability, enabling proactive decision-making in precision agriculture environments.

Future work may include:

- Integration of real weather forecast data
- Reinforcement Learning for automated irrigation optimization
- Deployment in real production greenhouse systems

References

Abdullah, W. (2024). *IoT Agriculture 2024*. Kaggle.com. https://www.kaggle.com/datasets/wisam1985/iot-agriculture-2024?

Beg, A., Kumar, M., & Kumar, M. (2025). IoT (INTERNET OF THINGS) IN AGRICULTURE. *ResearchGate*. https://www.researchgate.net/publication/398601888_IoT_INTERNET_OF_THINGS_IN_AGRICULTURE

Chen, X. (2025). The role of modern agricultural technologies in improving agricultural productivity and land use efficiency. *Frontiers in Plant Science*, *16*. https://doi.org/10.3389/fpls.2025.1675657

Deshpande, Y. D., & Rahman, S. R. (2023). Edge-Based Real-Time Sensor Data Processing for Anomaly Detection in Industrial IoT Applications. *Research Journal of Computer Systems and Engineering*, *4*(2), 16–30. https://doi.org/10.52710/rjcse.71

Eldho K J. (2025). Optimizing Data Transfer Speed and the Performance Evaluation of MQTT in IoT: A Study on MQTT for Atmospheric Condition Monitoring. *Journal of Information Systems Engineering and Management*, *10*(39s), 231–244. https://doi.org/10.52783/jisem.v10i39s.7144

Eze, V. H. U., Eze, E. C., Alaneme, G. U., BUBU, P. E., Nnadi, E. O. E., & Okon, M. B. (2025). Integrating IoT sensors and machine learning for sustainable precision agroecology: enhancing crop resilience and resource efficiency through data-driven strategies, challenges, and future prospects. *Discover Agriculture*, *3*(1). https://doi.org/10.1007/s44279-025-00247-y

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., & Zhang, H. (2019). Deep Learning with Long Short-Term Memory for Time Series Prediction. *IEEE Communications Magazine*, *57*(6), 114–119. https://doi.org/10.1109/mcom.2019.1800155

Liu, Y., Johar, M. G. M., & Hajamydeen, A. I. (2024). IoT-Based Real Time Greenhouse Monitoring and Controlling System. *ITEGAM- Journal of Engineering and Technology for Industrial Applications (ITEGAM-JETIA*, *10*(48), 01–07. https://doi.org/10.5935/jetia.v10i48.895

Michael, A., & William, E. (2025, February 20). *Machine Learning Algorithms for Demand Forecasting*. ResearchGate; unknown. https://www.researchgate.net/publication/389357099_Machine_Learning_Algorithms_for_Demand_Forecasting.

Appendix

Github Link: https://github.com/tanvisingh-11/Final-Project-AAI-530-IN1-Group-9

Tableau:https://public.tableau.com/authoring/Final_Project_AAI_530_IN1_Group_9/Dashboard1

Python Codes

```
# Data Preprocessing (Objective 1)

# Load the dataset

import pandas as pd

# Load your data (adjust the path for your dataset in Colab)

df = pd.read_excel('/content/IoT Agriculture 2024(1).xlsx')

# Display the first few rows of the dataset

df.head()

# Handling missing values (if any)

df = df.dropna()

# Convert date column to datetime

df['date'] = pd.to_datetime(df['date'])

# Encoding categorical columns (if needed, for instance, one-hot encoding)

df = pd.get_dummies(df, columns=['Fan_actuator_OFF', 'Fan_actuator_ON',

'Watering_plant_pump_OFF', 'Watering_plant_pump_ON',

'Water_pump_actuator_OFF', 'Water_pump_actuator_ON'], drop_first=True)

# Feature engineering: Select relevant features

features = ['temperature', 'humidity', 'water_level', 'N', 'P', 'K']

target = 'temperature'  # Example target for prediction

# Split data into training and testing sets

from sklearn.model_selection import train_test_split
```

```python
X = df[features]

y = df[target]

# Split into training and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

print(X_train, X_test, y_train, y_test)

# LSTM Model for Time Series Prediction (Objective 2)

# LSTM Model to predict temperature or other variables

from keras.models import Sequential

from keras.layers import LSTM, Dense

import numpy as np

# Prepare data for LSTM model (reshape data for time series)

X_train_reshaped = X_train.values.reshape((X_train.shape[0], 1, X_train.shape[1]))

X_test_reshaped = X_test.values.reshape((X_test.shape[0], 1, X_test.shape[1]))

# Build the LSTM model

model = Sequential()

print(model)

model.add(LSTM(50, activation='relu', input_shape=(X_train_reshaped.shape[1],

X_train_reshaped.shape[2])))

model.add(Dense(1))

model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model

model.fit(X_train_reshaped, y_train, epochs=20, batch_size=32,

validation_data=(X_test_reshaped, y_test))

# Predict and evaluate

y_pred = model.predict(X_test_reshaped)
```

```
print(y_pred)

# Evaluate model performance

from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(y_test, y_pred)

print(f"Mean Absolute Error: {mae}")

# Linear Regression for Comparison (Objective 2)

# Linear Regression Model for comparison

from sklearn.linear_model import LinearRegression

# Initialize and train the linear regression model

lr_model = LinearRegression()

print(lr_model)

lr_model.fit(X_train, y_train)

# Make predictions

y_lr_pred = lr_model.predict(X_test)

print(y_lr_pred)

# Evaluate linear regression performance

lr_mae = mean_absolute_error(y_test, y_lr_pred)

print(f"Linear Regression Mean Absolute Error: {lr_mae}")

# Data Visualization (Objective 1)

# Set the size of the figure

plt.figure(figsize=(12, 8))

# Plot temperature and humidity data

plt.plot(df['date'], df['temperature'], label='Temperature (°C)', color='tab:red',

linewidth=2)

plt.plot(df['date'], df['humidity'], label='Humidity (%)', color='tab:blue', linewidth=2)
```

```python
# Customize the plot with labels, title, and grid

plt.xlabel('Date', fontsize=14)

plt.ylabel('Value', fontsize=14)

plt.title('Temperature and Humidity Trends Over Time', fontsize=16, fontweight='bold')

# Rotate x-axis labels for better readability

plt.xticks(rotation=45, ha='right')

# Add a legend to differentiate between the two lines

plt.legend(loc='upper left', fontsize=12)

# Add gridlines to make it easier to read values

plt.grid(True, linestyle='--', alpha=0.7)

# Display the plot

plt.tight_layout()

plt.show()

# Evaluation of Cost Reduction, Sustainability, and Compliance (Objective 3)

# Measure the impact of predictions on sustainability metrics, cost reduction

# Assuming you have a function to calculate emissions or cost

# Calculate the predicted emissions reduction based on optimized supply chain

predicted_cost_savings = np.sum(y_pred) - np.sum(y_lr_pred)

print(f"Predicted Cost Savings: {predicted_cost_savings}")
```