Weather-Driven Agricultural Decision-Making Under Imperfect Conditions

Tamim Ahmed
tamim.ahmed@wsu.edu
Washington State University, USA

Monowar Hasan
monowar.hasan@wsu.edu
Washington State University, USA

ABSTRACT

Our research demonstrates the usefulness of data analytics using digital twin for detecting inconsistencies in weather measurements, which are critical to various agricultural decision-making and automation tasks. By leveraging digital twin technology, we develop a *modular* framework named CEREALIA that allows end-users to check for data inconsistencies when perfect weather feeds are unavailable. CEREALIA uses neural network models to check anomalies and aids end-users in informed decision-making. We develop a prototype of CEREALIA using the NVIDIA Jetson Orin platform and test it with an operational weather network established in a commercial orchard.

CCS CONCEPTS

• **Applied computing** → **Agriculture**.

KEYWORDS

Digital Twin, Agriculture Weather Networks

ACM Reference Format:
Tamim Ahmed and Monowar Hasan. 2025. Weather-Driven Agricultural Decision-Making Under Imperfect Conditions. In *The 33rd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '25), November 3–6, 2025, Minneapolis, MN, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3748636.3764158

1 INTRODUCTION

Modern agriculture, especially high-value specialty crop production, relies heavily on data-driven automation technologies for production management. Agricultural decision-making increasingly relies on precise and consistent environmental data to optimize crop health, resource utilization, and yield predictions. Regional as well as localized weather critically drives several on-farm decision-support models and intelligent automated systems, including irrigation, abiotic stress (e.g., cold, heat) mitigation, and pest management, among others [1]. Weather data measurements, which include essential metrics such as temperature, humidity, and precipitation, form the foundation of such decisions. However, weather networks, like any geo-distributed cyber-physical systems, are often plagued by *inconsistencies* due to sensor faults, calibration drift, communication errors, environmental interference, or cyber breaches. Imperfect measurements can significantly undermine the reliability of downstream applications in agriculture, ranging from predictive modeling to real-time interventions. Such discrepancies can lead to inaccurate predictions, suboptimal resource allocation, and reduced agricultural productivity. To address this problem, we adopt the *digital twin* technology [2] and develop a *modular* data analyzer for assessing and understanding weather feeds and their impact on growers' decision-making process. We name our framework CEREALIA.[1]

A key feature of CEREALIA is to design it as a *modular tool* that end-users can utilize for several in-farm decision-making tasks when perfect physical measurements are unavailable. We build dynamic virtual data streams that mirror physical stations—to monitor, validate, and enhance the reliability of weather data in the face of anomalies. While digital twin technology is used in other domains (e.g., control systems, manufacturing, robotics), its benefits are not fully explored for digital agriculture. By leveraging the concept of digital twins, we show that it is feasible to create a virtual replica of weather networks for real-time inconsistency detection, data imputation, and predictive analytics. As a user-inspired scenario, we study inconsistency implications, i.e., how erroneous weather data impacts in-farm decision-making. Specifically, our case study involves *predicting fruit surface temperature*, which is a crucial indicator of fruit stress. Growers use temperature prediction tools to initiate automated cooling precautions to reduce crop losses. We show that CEREALIA is useful to observe whether inconsistent data affects the fruit surface temperature prediction.

Our Contributions. This is an interdisciplinary research where the core is a spatially distributed system (weather network) used for domain-specific (viz., agriculture) data analytics. Although biosystems researchers develop models for agricultural decision-making tasks, they often assume a perfect setup in which sensor readings are free from faults or attacks—an assumption that does not hold in practice. Our work leverages the power of data analytics tools (viz., machine learning models) to bridge the gap between the physical weather network and data-driven agricultural decision-making in the presence of imperfect measurements. We note that time series prediction and anomaly detection are not new research areas. However, our contribution is *not* on proposing a new detection or prediction algorithm. Instead, we introduce a "plug-and-play" architecture that enables ag-tech stakeholders to: *(a)* use any learning tools to analyze inconsistencies in measurements and *(b)* evaluate how these inconsistencies impact decision-making

This research is partly supported by the U.S. National Science Foundation Award 2345653. Any findings, opinions, recommendations, or conclusions expressed in the paper are those of the authors and do not necessarily reflect the sponsor's views.

SIGSPATIAL '25, November 3–6, 2025, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2086-4/2025/11
https://doi.org/10.1145/3748636.3764158

[1]In ancient Roman religion, Cerealia was a major festival dedicated to the sowing, growing, and harvesting crops. As our work aims to assist in better decision-making for agricultural stakeholders, we name our framework CEREALIA.

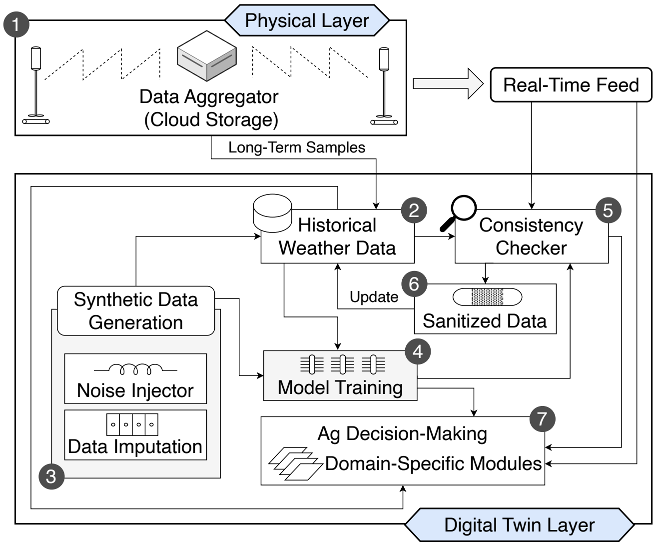

Figure 1: Workflow of Cerealia. We make Cerealia modular that allows designers to integrate various noisy data that can be used with historical weather traces to train machine learning model(s). A runtime consistency checker module uses a trained model to check for imperfect measurements and their impact on targeted decision-making applications (for instance, fruit heat/frost prediction).

activities (e.g., predicting the temperature of fruits). With a digital twin tool like Cerealia, end-users can analyze inconsistencies in weather data and their impact on agricultural decision-making in real time. Our modular design allows seamless integration of both new and existing machine learning and decision-making models.

Cerealia is implemented and tested by using a commercial orchard located in Central Washington, United States and maintained by Washington State University (§3–§4). The data analyzer modules (e.g., learning models) are deployed on an NVIDIA Jetson Orin platform [3]. We measure the overheads of running learning models on our evaluation platform and test the scalability of Cerealia. We find that the overheads of checking (in terms of timing and memory usage) are minimal (less than 1 s and 0.5 MB of memory use). Our technical report [4] presents additional evaluation results.

2 DESIGN AND WORKFLOW

The high-level architecture of Cerealia is illustrated in Fig. 1. We design Cerealia using a two-layer architecture: *(a) physical* layer and *(b) digital twin analyzer* layer, as we present below.

2.1 Components of Cerealia

2.1.1 Physical Layer. As shown in Block 1 of Fig. 1, the physical layer consists of weather stations deployed in the field. Many off-the-shelf weather stations (e.g., Cabled Vantage Pro2 [5], ATMOS 41W [6]) and existing weather networks (e.g., AgWeatherNet [7]) generally collect field measurements from stations in a periodic interval and store them in a cloud gateway for further processing. We follow a similar approach in Cerealia, where a cloud storage service collects data from distributed stations and feeds this to our analysis engines in the digital twin layer (see next).

2.1.2 Digital Twin Layer. The digital twin layer is the crux of the Cerealia and contains several modules (Blocks 2-7 in Fig. 1). In the *offline phase* (gray-shaded blocks in the data analyzer layer), Cerealia collects historical field measurements accumulated over the years from stations of interest or existing region-specific weather datasets and stores them in its internal database (Block 2 in the figure). These historical traces could be sanitized (i.e., free from anomalous data) or may have inconsistent measurements. As the focus of Cerealia is to emulate various inconsistent scenarios and see how they affect application-specific decision-making, we incorporate a noise generator module (Block 3) that allows us to inject various types of anomalous data. We train machine learning models based on historical observations and noise traces to check for inconsistent data at runtime (Block 4).

The *runtime* components of Cerealia (Blocks 5-7) use trained models and real-time measurements obtained from the physical layer to check for data inconsistencies. If anomalous data is detected, Cerealia notifies the user. Further, the historical database can be further updated by using sanitized or imputed data (Block 6) from the consistency checker (Block 5). Besides, the users can use the trained models and real-time measurements to see how noisy data affects the prediction of a given agricultural task (Block 7).

Cerealia supports *incremental* learning and updates. This allows Cerealia to adapt for the targeted weather network, which is often the key requirement for regional agricultural decision-making. For instance, as new samples are generated, Cerealia updates the historical database and retrains the model with new observations (the circular loops in Blocks 5, 6, 2, 4). As model training typically takes time, and weather data is generated at a higher volume/frequency, designers can opt for model retraining at a coarse granularity. The granularity of updating the historical database and the retraining models is left as a designer-chosen parameter. Our evaluation considers an on-shot scenario (i.e., without any retraining) and demonstrates the performance of inference and effectiveness of runtime decision-making, as model training is typically conducted offline (i.e., when the system is not operational).

The key components of the data analyzer layer (i.e., Blocks 3-7) are modular by design. Our current implementation incorporates four kinds of noisy data and uses nine machine-learning models to see the models' behavior under these inconsistencies to find the best possible solution for Cerealia. However, other faulty/anomalous data and other statistical models or anomaly detection tools can be incorporated with Cerealia to check for inconsistencies. For demonstration purposes, we tested Cerealia for an agricultural use cases (e.g., fruit surface temperature prediction problem).[2] However, this component of Cerealia (i.e., Block 7) can be adapted for any other agriculture decision-making tasks (e.g., fruit bloom phenology [8], soil water content prediction [9]) without loss of generality.

[2]Additional use cases presented in our technical report [4].

2.2 Decision-Making under Imperfections

At each time instance t, each station captures a set of readings from n sensors. Let us represent these readings as follows: $\mathbf{X}(t) = \{x_1(t), x_2(t), \ldots, x_n(t)\}$, where $x_i(t)$ represents the recorded value for weather attribute i (e.g., air temperature, atmospheric pressure, solar radiation, wind speed). As mentioned in §2.1, Cerealia maintains a database of historical weather traces. Let us denote the historical dataset as follows: $H = \{\mathbf{X}(t_1), \mathbf{X}(t_2), \ldots, \mathbf{X}(t_m)\}$, where $t_1, t_2, \ldots, t_m$ are the timestamps of the recorded entries. Note that this historical dataset H does not necessarily contain all sanitized or perfect measurements. For a given time instance t, we mimic imperfect environments during model training by injecting noisy samples, $\mathbf{X}_{noise}(t)$.

Cerealia allows the end-users to use the trained models (obtained from historical observations H, including inconsistent samples) and make informed decisions about their target applications. For instance, we can run an inference operation to check for inconsistent weather data at any time epoch t. Cerealia continuously fetches data from the physical layer and then classifies incoming sequences as clean or anomalous. Given a new real-time sequence of size k i.e., $\mathbf{X} = \{X(t-k), \ldots, X(t)\}$, the model classifies it to detect any anomalies as follows: $y_{\text{real-time}} = f_{\text{pretrained\_model}}(\mathbf{X})$, where $f_{\text{pretrained\_model}}(\mathbf{X})$ is the output (inference) from the machine learning model. If $y_{\text{real-time}} \neq$ `clean`, the sequence is flagged as inconsistent, indicating an anomaly. In our current implementation, any data labeled as anomalous triggers an alert for the operator.[3]

3 IMPLEMENTATION

We integrate Cerealia with a commercial apple orchard located in Quincy, Washington, United States (47°13′31.3″ 119°57′38.0″W). The orchard has an average elevation of 322 m and an average slope of 2° facing east. The station where we have access is installed at 1.5 m above ground level. The station uses an all-in-one weather sensor (ATMOS 41 from METER Group [6]). It measures several weather parameters such as solar radiation, precipitation, vapor pressure, relative humidity, barometric pressure, horizontal wind speed, wind gust, wind direction, tilt, lightning strike, and lightning average distance. Weather stations sample measurements at 5-minute intervals and send the data to a vendor cloud service (ZENTRA Cloud [10]). We obtained weather data from March 2023 to January 2025. The collection frequency is twelve data tuples per hour per station (recall: stations send data to the cloud in 5-minute intervals), resulting in 192,613 data points.

We deploy the data analyzer layer on NVIDIA Jetson Orin Developer Kit (12-core Arm Cortex-A78AE 64-bit CPU, 64 GB LPDDR5 RAM, and 2048-core GPU with 64 Tensor Cores) [3]. The algorithms are implemented in Python and executed on Linux (kernel version 5.15.148). We crawl weather station data from ZENTRA Cloud through HTTP requests using the Python `requests` package. We use TensorFlow [11] for model training and inference purposes.

[3]Other response measures are possible (although not incorporated), for instance, ignoring the measurements from the station or sending a command to the physical layer to turn off or reset the station.

Cerealia implementation is **publicly available**: https://github.com/CPS2RL/ag-dt. Additional implementation details and evaluation results are available in our technical report [4].

4 CASE STUDY: FRUIT SURFACE TEMPERATURE PREDICTION

Heat stress to maturing fruit is a key concern for tree fruit growers. Localized weather-based fruit surface temperature (an indicator of fruit stress) can help in planning better mitigation strategies and ultimately reduce crop losses. To reduce fruit stress, growers use overhead cyclic rotating sprinklers, foggers, netting, and protectant sprays [12]. However, precise data inputs are needed for the actuation and effective use of some of these mitigation techniques. Faulty or incorrect weather estimations can affect the initiation of such protective measures. We now show how Cerealia could be useful to predict fruit surface temperature so that growers can make informed decisions.

We use nine learning models (see Table 1, additional model details are provided in our technical report [4]) to predict the apple surface temperatures from the weather measurements obtained from the orchard. The weather parameters for surface temperature prediction include canopy air temperature (°C), wind speed (m/s), dew point (°C), and solar radiation (W/m^2). Past research shows fruit surface temperature can be estimated from these weather attributes [13]. We study for both perfect and imperfect weather feeds. For imperfect cases, approximately 20% of the weather sensor feeds are faulty/inconsistent samples, and like before, we use equal noise splits (i.e., 5% for each class).

In the absence of faulty data (i.e., column labeled with "No Imperfection" in Table 1), CNN and Transformer-based models (in particular, ResNet and Informer) show best performance with the lowest MAE, RMSE, and R^2 numbers. Based on these findings, we conclude that convolutional and attention-based models perform well on clean (consistent) weather data. We then check for cases where we have noisy readings (the "Imperfect Measurement" column in Table 1). As Table 1 shows, the performance of the models (MAE, RMSE, and R^2) drops in the presence of inconsistent measurements. If we perform predictions ignoring faulty measurements, it can lead to incorrect decision-making for farmers who rely on the apple surface temperature values to initiate proper protective actions (such as adjustment of irrigation or covering fruits). Hence, we also test how data analytics can assist in this case. The last column in Table 1 ("Imputing Inconsistencies") shows the performance numbers when Cerealia replaces faulty measurements with imputed values using a generative model (e.g., C-RNN-GAN [14]). As the table shows, imputing faulty samples with the expected value improves overall prediction performance (i.e., the errors are lower than in the imperfect case). For this prediction problem, our experiments show GRU, TST, and TST-AE have more variability (higher MAE and RMSE but lower R^2). We attribute this to the model's internal architectures, which cause them to learn less efficiently (larger errors). ResNet and Informer models show better results for predicting the surface temperature in noisy measurements. This case study further indicates how growers can leverage the modularity of Cerealia and customize it for weather-driven decision-making problems.

Table 1: Using CEREALIA to predict apple surface temperature.

	No Imperfection			Imperfect Measurements			Imputing Inconsistencies		
Models	**MAE**	**RMSE**	**R^2**	**MAE**	**RMSE**	**R^2**	**MAE**	**RMSE**	**R^2**
TCN	0.6874	1.2911	0.9288	1.9347	4.9019	0.2634	0.7652	1.3466	0.9226
ResNet	0.5823	1.2395	0.9344	2.1711	5.4140	0.1014	0.6695	1.3038	0.9274
LSTM	0.8335	1.3969	0.9167	1.9208	4.1776	0.4650	0.9215	1.4688	0.9079
Bi-LSTM	1.0283	1.5890	0.8922	1.9139	3.8689	0.5411	1.0959	1.6467	0.8842
GRU	1.7242	2.1495	0.8027	2.3225	3.8069	0.5557	1.8041	2.2292	0.7879
TST	0.8128	1.3729	0.9195	2.4770	5.3600	0.1193	0.9348	1.8015	0.8615
Informer	0.5983	1.2530	0.9330	2.2968	5.3474	0.1234	0.6769	1.3122	0.9265
TST-AE	1.5949	1.9784	0.8329	2.3072	3.5205	0.6201	1.6834	2.0721	0.8167
LSTM-AE	0.9393	1.4239	0.9134	1.7415	3.4767	0.6295	1.0146	1.4949	0.9046

5 RELATED WORK AND CONCLUSION

There has been some work in weather-driven digital agriculture that leverages machine learning models for several applications such as irrigation management, disease modeling, crop yield estimation, fruit stress estimation, and real-time farming decision making [15–24]. While such research facilitates digital agriculture, data quality and consistency challenges are not explored. Fault handling is widely investigated [25–28], but their consequences in weather applications are not well explored. Researchers also explore the weather data imputation problem [1, 29]. However, unlike our comprehensive design space exploration, prior study is limited to assessing only a single statistical or machine learning model.

In contrast to existing research, we introduce a novel use of spatial data analysis models to assess data inconsistency issues in agricultural weather networks. To our knowledge, CEREALIA is one of the first comprehensive digital twin tools for resiliency analysis of weather-driven agricultural systems. CEREALIA will be a fundamental tool for twinning physical entities to provide in silico emulation capabilities and deliver insights to farm decision-making.

REFERENCES

[1] J. P. Boomgard-Zagrodnik and D. J. Brown, "Machine learning imputation of missing mesonet temperature observations," *Computers and Electronics in Agriculture*, vol. 192, p. 106580, 2022.

[2] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE access*, vol. 7, pp. 167653–167671, 2019.

[3] L. S. Karumbunathan, "NVIDIA Jetson AGX Orin series." https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf, 2022. Accessed: 03-27-2025.

[4] T. Ahmed and M. Hasan, "Weather-driven agricultural decision-making using digital twins under imperfect conditions," *arXiv preprint arXiv:2508.08326*, 2025.

[5] "Cabled Vantage Pro2 Plus Sensor Suite Weather Station." https://www.davisinstruments.com/products/cabled-vantage-pro2-plus-iss. [Accessed 03-12-2024].

[6] "ATMOS 41." https://metergroup.com/products/atmos-41/. [Accessed 03-12-2024].

[7] "Washington State University AgWeatherNet (AWN)." https://weather.wsu.edu. [Accessed 03-12-2024].

[8] B. Chaves, M. Salazar, T. Schmidt, N. Dasgupta, and G. Hoogenboom, "Modeling apple bloom phenology," in *X International Symposium on Modelling in Fruit Research and Orchard Management 1160*, pp. 201–206, 2015.

[9] K. Nath, P. Nayak, and K. Kasiviswanathan, "Soil volumetric water content prediction using unique hybrid deep learning algorithm," *Neural Computing and Applications*, pp. 1–23, 2024.

[10] "ZENTRA Cloud." https://metergroup.com/platform/zentra-cloud/. [Accessed 04-12-2024].

[11] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "TensorFlow: a system for large-scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

[12] J. Racsko and L. Schrader, "Sunburn of apple fruit: Historical background, recent advances and future perspectives," *Critical reviews in plant sciences*, vol. 31, no. 6, pp. 455–504, 2012.

[13] L. Li, T. Peters, Q. Zhang, J. Zhang, and D. Huang, "Modeling apple surface temperature dynamics based on weather data," *Sensors*, vol. 14, no. 11, pp. 20217–20234, 2014.

[14] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[15] F. Pierce and T. Elliott, "Regional and on-farm wireless sensor networks for agricultural systems in eastern washington," *Computers and electronics in agriculture*, vol. 61, no. 1, pp. 32–43, 2008.

[16] S. A. Kumar and P. Ilango, "The impact of wireless sensor network in the field of precision agriculture: A review," *Wireless Personal Communications*, vol. 98, pp. 685–698, 2018.

[17] S. Dandrifosse, A. Jago, J. P. Huart, V. Michaud, V. Planchon, and D. Rosillon, "Automatic quality control of weather data for timely decisions in agriculture," *Smart Agricultural Technology*, vol. 8, p. 100445, 2024.

[18] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station," *Soft Computing*, vol. 24, pp. 16453–16482, 2020.

[19] M. A. Zaytar and C. El Amrani, "Sequence to sequence weather forecasting with long short-term memory recurrent neural networks," *International Journal of Computer Applications*, vol. 143, no. 11, pp. 7–11, 2016.

[20] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 11121–11128, 2023.

[21] S. Khaki and L. Wang, "Crop yield prediction using deep neural networks," *Frontiers in plant science*, vol. 10, p. 621, 2019.

[22] B. King and K. Shellie, "Evaluation of neural network modeling to predict non-water-stressed leaf temperature in wine grape for calculation of crop water stress index," *Agricultural water management*, vol. 167, pp. 38–52, 2016.

[23] X. A. Wang, J. Tang, and M. Whitty, "Deepphenology: Estimation of apple flower phenology distributions based on deep learning," *Computers and Electronics in Agriculture*, vol. 185, p. 106123, 2021.

[24] R. Ranjan, L. R. Khot, R. T. Peters, M. R. Salazar-Gutierrez, and G. Shi, "In-field crop physiology sensing aided real-time apple fruit surface temperature monitoring for sunburn prediction," *Computers and Electronics in Agriculture*, vol. 175, p. 105558, 2020.

[25] Z. Noshad, N. Javaid, T. Saba, Z. Wadud, M. Q. Saleem, M. E. Alzahrani, and O. E. Sheta, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, p. 1568, 2019.

[26] T. Luo and S. G. Nagarajan, "Distributed anomaly detection using autoencoder neural networks in wsn for iot," in *2018 ieee international conference on communications (icc)*, pp. 1–6, IEEE, 2018.

[27] C. Tang, L. Xu, B. Yang, Y. Tang, and D. Zhao, "Gru-based interpretable multivariate time series anomaly detection in industrial control system," *Computers & Security*, vol. 127, p. 103094, 2023.

[28] Y. He and J. Zhao, "Temporal convolutional networks for anomaly detection in time series," in *Journal of Physics: Conference Series*, vol. 1213, p. 042050, IOP Publishing, 2019.

[29] I. G. Doreswamy and B. Manjunatha, "Performance evaluation of predictive models for missing data imputation in weather data," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1327–1334, 2017.