A Deployment-Oriented Benchmark of Machine Learning Models for Agroclimatic Forecasting Under Degraded Sensor Data

Oleksandr Zhabko , Ivan Laktionov * , Grygorii Diachenko , Oleksandr Vinyukov , Dmytro Moroz

Posted Date: 7 May 2026

doi: 10.20944/preprints202605.0372.v1

Keywords: agriculture; time-series forecasting; machine learning; fog computing; edge computing; IoT; environmental monitoring

 doi:10.20944/preprints202605.0372.v1

Article

A Deployment-Oriented Benchmark of Machine Learning Models for Agroclimatic Forecasting Under Degraded Sensor Data

Oleksandr Zhabko [1], Ivan Laktionov [2,3,*], Grygorii Diachenko [4], Oleksandr Vinyukov [5] and Dmytro Moroz [3]

[1] Department of Information Technology and Computer Engineering, Faculty of Information Technologies, Dnipro University of Technology, UA49005 Dnipro, Ukraine
[2] The SAN University, str. Sienkiewicza, 9, 90-113 Łódź, Poland
[3] Department of Software of Computer Systems, Faculty of Information Technologies, Dnipro University of Technology, UA49005 Dnipro, Ukraine
[4] Department of Electric Drive, Faculty of Electrical Engineering, Dnipro University of Technology, UA49005 Dnipro, Ukraine
[5] Donetsk State Agricultural Science Station of the National Academy of Agrarian Sciences of Ukraine, st. Zakhisnykiv Ukrainy, 1, UA85307 Pokrovsk, Donetsk Region, Ukraine
* Correspondence: laktionov.i.s@nmu.one

Abstract

This study evaluates the performance of classical machine learning models for one-step-ahead agroclimatic time-series forecasting under degraded sensor-data conditions. The motivation is the operation of IoT-based field monitoring systems, where measurements may be noisy, incomplete, temporally irregular, and constrained by limited local storage and computational resources. Using a real meteorological dataset collected by a field weather station in the Dnipro region of Ukraine, we compared twelve regression models: Ridge Regression, Random Forest, Extra Trees, Gradient Boosting, HistGradientBoosting, Support Vector Regression, Linear SVR, KNN, PLSRegression, ElasticNet, Lasso, and MultiTaskElasticNet. The models were evaluated under five controlled experimental scenarios: baseline data, missing values, additive noise, reduced training history, and combined noise–missingness degradation. The results indicate that Ridge Regression provides the highest accuracy under clean and mildly degraded conditions, whereas HistGradientBoosting is more stable under severe combined degradation. These findings support the use of deployment-oriented model selection for agroclimatic forecasting; however, the proposed edge/fog workflow should be interpreted as a conceptual deployment direction unless validated by direct measurements of latency, memory footprint, and energy consumption on representative hardware. The study therefore provides a benchmark for robustness-oriented model selection rather than a fully validated embedded deployment framework.

Keywords: agriculture; time-series forecasting; machine learning; fog computing; edge computing; IoT; environmental monitoring

1. Introduction

1.1. Relevance of the Topic and Research Motivation

The rapid development of the Internet of Things (IoT), wireless sensor networks, and distributed computing has driven a shift from centralized cloud analytics toward decentralized fog and/or edge architectures. Unlike the traditional "from sensor to cloud" paradigm, in which all data must be transmitted to a remote data center, fog and/or edge computing enables data processing directly on peripheral devices, from sensors to gateways, microcontrollers, or local edge nodes [1,2]. This

 doi:10.20944/preprints202605.0372.v1

approach reduces latency, decreases network traffic, improves resilience to connectivity loss, and ensures autonomous system operation.

The need for local analytics is particularly crucial in precision agriculture, where real-time decision-making has a direct impact on crop health and yield. Edge-level processing plays a key role in tasks such as disease detection, pest monitoring, forecasting the development of plant diseases, evaluating microclimatic risks, and planning agrotechnical interventions. In these environments, even minimal delays in data transmission or processing may lead to crop loss or ineffective implementation of field management measures [3].

Forecasting environmental time series, including temperature, humidity, atmospheric pressure, wind speed, dew point, and related parameters, is a key task in these systems. However, real-world IoT sensor data often contain missing values, noise, anomalies, and irregular timestamps, which significantly complicate modeling [4]. An additional challenge arises from the hardware limitations of fog and/or edge devices, including restricted memory, energy consumption, and computational capacity, which make complex or unstable models unsuitable for deployment [5].

Classical statistical approaches such as ARIMA and SARIMA remain popular due to their simplicity, interpretability, and low computational costs. Nevertheless, their accuracy declines sharply under non-stationarity, noise, or missing data, conditions typical for sensor networks [6]. Furthermore, these models require frequent retraining as new observations arrive, which limits their applicability on resource-constrained fog and/or edge devices that operate in real-time [7].

Hybrid methods, such as SARIMA+LSTM or SARIMA+XGBoost, often report improvements of 10–25% compared to purely statistical approaches [8]. However, they exhibit significantly higher computational complexity, increased memory consumption, and long training times, making them impractical for direct deployment on edge devices [9].

Against this background, machine learning (ML) models, including regularized linear methods, support-vector regressors, instance-based learners, and ensemble-based approaches, have become widely used for environmental time-series forecasting. These models can be trained offline and then deployed for inference on peripheral devices, provided that their memory footprint, latency, and robustness to degraded input data are compatible with the target edge or fog platform [10,11]. Contemporary research confirms that ML-based approaches are the most promising solutions for time-series forecasting under fog and/or edge computing constraints, where accuracy, speed, and computational efficiency must be balanced [12].

This study focuses on classical ML regressors rather than on statistical, hybrid, or deep learning models. This choice was made to keep the experimental scope centered on models that are commonly available in lightweight, Python-based inference pipelines and that can be evaluated consistently under the same degradation protocol. However, this restriction also limits the comparative scope of the study. Therefore, the results should not be interpreted as evidence of the general superiority of classical ML over ARIMA/SARIMA, Prophet, hybrid models, or lightweight neural architectures. Instead, the study provides an internal benchmark of selected classical ML models under controlled degradation scenarios, while broader cross-family comparisons remain a direction for future work.

1.2. Analysis of the Latest Research and Publications

Recent surveys on environmental and weather-related time-series forecasting provide a structured overview of existing modeling approaches, ranging from classical statistical techniques to hybrid and deep learning models [6,12]. Within this landscape, ARIMA, SARIMA, and Holt–Winters remain widely used as baseline forecasting tools due to their interpretability and computational efficiency. However, comparative analyses show that these models are typically evaluated under controlled conditions and tend to underperform when applied to real-world IoT data, which are characterized by irregular sampling, sensor noise, or missing observations [4,7]. Importantly, prior work rarely accounts for the deployment constraints of fog and/or edge devices, which limits the practical applicability of statistical models beyond conventional server-side environments.

 doi:10.20944/preprints202605.0372.v1

A second major line of research investigates hybrid models such as SARIMA+LSTM, SARIMA+XGBoost, STL-XGBoost, and Wavelet-LSTM. These architectures demonstrate accuracy improvements of 10–25% over purely statistical baselines across various environmental forecasting tasks [8,12]. Nevertheless, existing analyses consistently highlight that hybrid approaches introduce considerable computational overhead, require substantial memory allocations, and involve long training cycles, making them impractical for hardware-constrained fog and/or edge platforms [9]. As a result, most hybrid solutions are designed for cloud-based inference pipelines rather than for fully decentralized IoT ecosystems.

Another important research direction concerns noise and missing sensor data, which are inherent issues in meteorological and agricultural IoT systems. Systematic reviews estimate that 15–30% of measurements in real deployments may be missing or corrupted [4]. Common imputation strategies include forward/backfill, KNN-based methods, spline or Lagrange interpolation, and ML-based approaches such as MICE or Autoencoders. Empirical evidence indicates that tree-based models (Extra Trees, Random Forest) are generally more resilient to incomplete or noisy data than ARIMA or LSTM due to their ensemble structure and intrinsic randomness [8,10].

In contrast to statistical and hybrid methods, ML approaches have become dominant in IoT and fog/edge time-series forecasting. Ensemble models, such as Random Forest, Extra Trees, Gradient Boosting, and HistGradientBoosting, demonstrate high predictive accuracy and maintain stable inference times (50–100 ms on microcomputer hardware, such as Raspberry Pi) when trained in the cloud and deployed on peripheral nodes [11].

Neural network architectures, including LSTM, GRU, and 1D-CNN, achieve even higher accuracy in handling highly nonlinear and long-range temporal dependencies. However, they require significantly more memory and computational power, often exceeding the capabilities of fog and/or edge hardware unless optimized through quantization, pruning, or deployment via TensorFlow Lite or ONNX Runtime. Although deep neural networks are extensively investigated in the literature, the present study focuses exclusively on classical ML models due to fog and/or edge constraints [11,13].

A more recent trend is TinyML, which enables inference directly on microcontrollers with ≤256 KB RAM and power budgets ≤1 W. Studies [14,15] demonstrate that micro-LSTM, compact GRU models, and lightweight gradient boosting variants can run entirely offline on embedded devices, enabling fully autonomous IoT systems.

Recent agricultural studies further demonstrate the applicability of ML-based forecasting under real field conditions. Research on Fusarium head blight prediction in corn [16] shows that ensemble learning and gradient boosting methods provide high forecasting accuracy when driven by local meteorological variables. Another study [17] presents an IoT-oriented decision-making network for evaluating the probability of crop diseases, highlighting the importance of distributed sensing and edge-level analytics in agricultural monitoring systems. A more advanced decision-support framework [18] integrates weather monitoring, ML, and explainable artificial intelligence to estimate the risk of wheat powdery mildew, illustrating how environmental sensing can be combined with ML models for operational agronomic decision-making.

Overall, contemporary literature consistently highlights the advantages of ML-based forecasting over statistical and hybrid models in fog and/or edge computing contexts. Among these, Ridge Regression, HistGradientBoosting, Gradient Boosting, and Random Forest are considered the most balanced in terms of accuracy, robustness to noise and missing data, and computational efficiency [10,12]. Modern gradient boosting frameworks, such as LightGBM, further demonstrate high efficiency and scalability for large-scale and degraded datasets [19], while ensemble learning strategies based on locally independent predictors improve generalization and robustness under uncertainty [20]. Additionally, recent studies on adaptive noise suppression in sensor systems confirm the key role of robust signal processing for maintaining predictive stability under real-world interference conditions [21]. This aligns with the practical requirements of resource-constrained

 doi:10.20944/preprints202605.0372.v1

devices, where inference speed, robustness, and model stability are as important as predictive accuracy.

1.3. Aim, Objectives, Object and Subject of the Study

Despite extensive research on forecasting meteorological time-series data, the existing literature still lacks a comprehensive evaluation of ML models under the conditions typical for fog and/or edge computing environments. Most prior studies focus either on improving forecasting accuracy or on optimizing specific algorithms, while paying insufficient attention to the operational constraints and data degradation patterns inherent to real IoT systems.

Classical statistical approaches, such as ARIMA and SARIMA, remain widely used because of their simplicity and interpretability. However, they exhibit strong performance only under strict assumptions of stationarity, low noise levels, and complete historical records. In the presence of missing data, stochastic disturbances, or even minor structural shifts in seasonality, their accuracy decreases sharply, and the models require retuning or full retraining. These limitations restrict the applicability of classical statistical models for real-time peripheral deployment.

Despite higher accuracy, hybrid models remain poorly suited for fog and/or edge deployment due to their computational and energy demands. Even when training is offloaded to the cloud, executing such models on peripheral nodes typically requires quantization, pruning, or hardware acceleration, which reduces their general applicability.

ML models, including Ridge Regression, Support Vector Regression, Random Forest, Extra Trees, Gradient Boosting, HistGradientBoosting, and Multilayer Perceptron, have recently demonstrated superior accuracy and robustness. However, most comparative studies evaluate models only on clean datasets, using metrics such as MAE, RMSE, and R^2. There is a noticeable lack of systematic analyses addressing how these models behave under realistic conditions involving sensor noise, missing observations, reduced historical data, or combined forms of degradation. Furthermore, there is little evidence regarding which algorithms remain operationally viable on devices with constrained memory and computational capacity, such as IoT gateways and edge nodes.

The research gap addressed in this study is the limited availability of deployment-oriented benchmarks that evaluate classical ML models under multiple forms of sensor-data degradation using a consistent time-series forecasting protocol. Prior work often emphasizes either predictive accuracy on clean datasets or algorithm-specific improvements, whereas fewer studies jointly examine how model rankings change under missing values, additive noise, reduced historical context, and combined degradation. In this study, the term “framework” refers to a structured experimental evaluation pipeline rather than a newly proposed learning algorithm. The contribution is therefore methodological and empirical: a consistent benchmark for assessing the accuracy–robustness trade-off of selected ML regressors in an agroclimatic IoT setting.

Based on the identified research gaps, the present study aims to provide a systematic experimental comparison of ML algorithms for forecasting environmental time-series parameters (such as temperature, humidity, and others) under fog and/or edge constraints. The objectives of this research are as follows:

1. To evaluate the accuracy of selected classical ML models under baseline, non-degraded data conditions.
2. To assess the robustness of these models under missing values, additive noise, reduced training history, and combined degradation.
3. To analyze the relative deployment suitability of the models from the perspective of algorithmic complexity and stability, without claiming full hardware-level validation.
4. To identify candidate models that provide a favorable accuracy–robustness trade-off for future edge/fog implementation in agroclimatic monitoring systems.

This formulation defines the scientific relevance of the study and provides the foundation for the experimental methodology described in the next section.

 doi:10.20944/preprints202605.0372.v1

2. Materials and Methods

This study utilizes real meteorological time-series data collected from a field weather station and stored in CSV format. The dataset contains eight meteorological variables: air temperature (°C), relative humidity (%), atmospheric pressure (hPa), wind speed (m/s), wind direction (degrees), precipitation intensity (mm/h), daily precipitation (mm/day), and dew point (°C). These variables are used consistently throughout the study as the monitored agroclimatic parameters. All timestamps were converted into the DateTime format and used as the index to ensure correct temporal ordering and consistency throughout the sequence.

The meteorological data were collected at a field monitoring site located in the Samar district of the Dnipro region, Ukraine. The measurements were obtained using a custom-built experimental weather station designed in accordance with the edge/fog computing architecture (Figure 1). The system is based on a serial microcontroller platform (Arduino Mega 2560) equipped with compatible meteorological sensors and a GSM/GPRS module for data transmission. Most preprocessing and computational procedures, including signal filtering, data validation, and the formation of structured measurement packets, are executed directly on the microcontroller. This device performs local buffering and fog-level relaying before transmitting the processed data to an IoT server deployed on a Raspberry Pi microcomputer, utilizing an information technology stack. The monitoring campaign spanned from May 10, 2023, to September 12, 2024, providing approximately 490 days of continuous observations with occasional gaps inherent to field IoT deployments.

Table 1. Dataset characteristics.

Parameter	Value
Total duration	from May 10, 2023 to September 12, 2024
Number of samples (raw)	58,802
Sampling interval	10 minutes (nominal)
Number of meteorological variables	8
Missing segments (timeline gaps)	Yes (several hours–days)
Missing values inside recorded segments	0% (sensor offline periods appear as gaps rather than NaNs)
Anomalous measurements removed	Yes (physically implausible humidity, temperature, wind spikes)

To provide a clear overview of the collected dataset, Figure 2 presents an eight-panel visualization, where each meteorological variable is shown on its own axis with its natural physical scale. This representation avoids the distortions of a single combined plot and allows each parameter: air temperature, relative humidity, atmospheric pressure, wind speed, wind direction, precipitation intensity, daily precipitation, and dew point, to be interpreted independently. The selected segment (November 2023 – February 2024) is continuous and stable, allowing for a clean inspection of short-term dynamics and sensor behavior relevant to the forecasting task.

The raw dataset consisted of N = 58,802 time-stamped observations collected at a nominal sampling interval of $\Delta t \approx 10$ minutes. The cleaned dataset contained no NaN values within the recorded measurement rows. However, the original monitoring timeline included natural timestamp gaps caused by field-level interruptions such as communication failures, power instability, or sensor downtime. Therefore, “no missing values” refers only to the absence of missing entries inside

 doi:10.20944/preprints202605.0372.v1

retained records, not to the absence of temporal gaps in the complete observation timeline. This distinction is important because the synthetic degradation experiments introduced controlled missingness into the retained data matrix, whereas naturally occurring timestamp gaps were treated as part of the field acquisition context. Figure 2 focuses on a continuous stable segment; the full dataset still contains natural timestamp gaps caused by field-level operational interruptions.

Figure 1. Acquisition equipment.

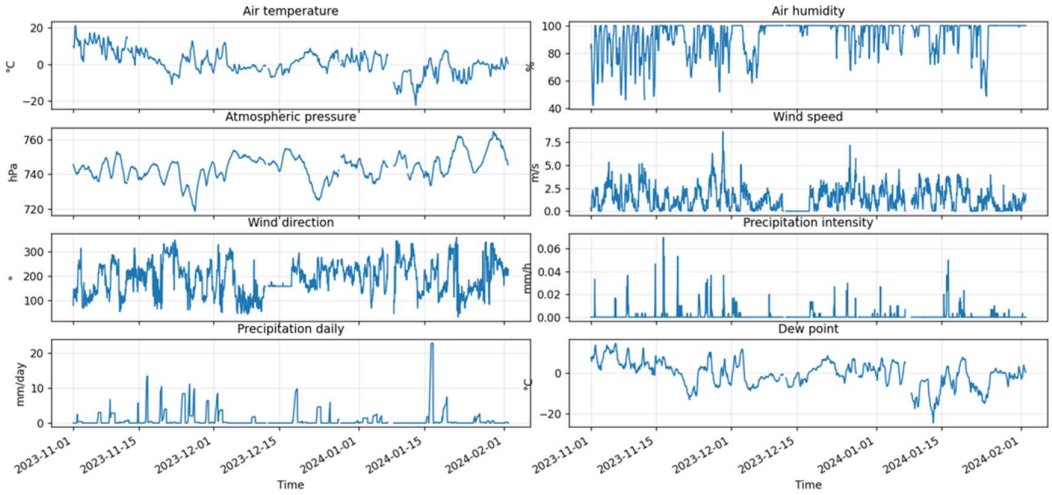


Figure 2. Time series of all meteorological variables collected by the edge-enabled field weather station during the monitoring.

The plots use the following axis labels and units: air temperature (°C), air humidity (%), atmospheric pressure (hPa), wind speed (m/s), wind direction (° – angle), precipitation intensity (mm/h), daily precipitation (mm/day), and dew point (°C).

Initial data cleaning involved removing duplicated measurements and discarding physically implausible values, such as negative relative humidity or temperature values outside realistic physical limits. These filtering criteria follow standard environmental data quality-control procedures and are consistent with the operational specifications of low-cost meteorological sensors. As IoT weather stations frequently produce irregular, noisy, or partially corrupted observations due to power interruptions and communication instability, this preprocessing step ensured a coherent and reliable foundation for subsequent model training.

Specifically, measurements were filtered using rule-based thresholds derived from the physical constraints of atmospheric parameters and manufacturer guidelines for field-grade sensing devices:

 doi:10.20944/preprints202605.0372.v1

humidity values below 0% or above 100% were removed; temperature readings below −40°C or above 60°C were discarded as physically implausible; negative wind-speed values were excluded; and abrupt, isolated single-point spikes, typically arising from transient electrical interference, were smoothed using a three-point median window. These corrections address common artefacts observed in embedded IoT systems and preserve the physical interpretability of the dataset.

To enable compatibility with supervised machine-learning models, the data were transformed into a lag-embedded representation. For each time series, lagged features were generated for the preceding six observation intervals:

$$(x_{t-1}, x_{t-2}, \dots, x_{t-6}) \qquad (1)$$

where x_t is the value of a meteorological variable at time t, x_{t-k} is its value at lag k, and $k \in \{1, \dots, 6\}$ defines the one-hour temporal window capturing short-term dependencies essential for forecasting.

Capturing short-term temporal dependencies is essential for one-step-ahead forecasting. The choice of six lags was guided by empirical inspection of the autocorrelation (ACF) and partial autocorrelation (PACF) functions, both of which indicated that the strongest dependencies are concentrated within the preceding hour (≈ 60 minutes). This lag depth offers a favorable trade-off between predictive accuracy and model complexity: adding additional lags did not significantly improve performance but increased memory requirements and inference latency, both of which are undesirable for resource-constrained fog and/or edge environments.

In addition to lagged values, causal rolling descriptors were computed to encode short-term variability and local atmospheric dynamics. To prevent information leakage, all moving averages and moving standard deviations were calculated only from observations available before the prediction instant. In implementation terms, the time series was shifted by one step before applying rolling-window operations. Thus, for a forecast issued at time t, the feature vector used observations up to $t-1$, while the target corresponded to the next observation. No centered rolling windows or future values were used.

Rolling statistics were computed over windows of 3, 6, and 12 previous observations, corresponding approximately to 30, 60, and 120 minutes under the nominal 10-minute sampling interval. Near the beginning of each temporal block, samples with insufficient history were removed rather than filled using future information. This preserves the causal structure required for realistic edge inference.

Moving averages capture local trends and smooth small-scale oscillations, whereas moving standard deviations quantify short-term variability and turbulence-like fluctuations. For each variable, these operations generated a set of temporal features that summarized the local structure of the signal.

By combining lagged observations and local statistical descriptors, the input feature vector for time t takes the form:

$$z_t = [x_{t-1}, x_{t-2}, \dots, x_{t-L}, \mathrm{MA}_3(x_t), \mathrm{MA}_6(x_t), \mathrm{MA}_{12}(x_t), \mathrm{SD}_3(x_t), \mathrm{SD}_6(x_t), \mathrm{SD}_{12}(x_t)] \qquad (2)$$

where z_t is the feature vector at time t, L is the number of lagged values, $\mathrm{MA}_w(x_t)$ and $\mathrm{SD}_w(x_t)$ are the moving average and moving standard deviation computed over a window $w \in \{3, 6, 12\}$, and each transformation is applied independently to every meteorological variable.

For d meteorological variables, the resulting feature vector has dimensionality:

$$m = d \cdot L + 6d \qquad (3)$$

where m is the dimensionality of the feature vector, d is the number of meteorological variables, L is the number of lag features per variable, and $6d$ corresponds to the statistical descriptors added for each variable.

The one-step-ahead forecasting task is defined as predicting the next multivariate observation:

$$y_t = x_{t+1} \qquad (4)$$

where y_t denotes the next-step target value and x_{t+1} is the true observation one time step ahead, which is a standard formulation for real-time predictive control in edge computing, where forecasts must be updated continuously at the device level.

A machine-learning model:

$$f_\theta : \mathbb{R}^m \to \mathbb{R}^d \qquad (5)$$

 doi:10.20944/preprints202605.0372.v1

where f_θ is a machine-learning model parameterized by θ, $\mathbb{R}^m$ is the input feature space, and $\mathbb{R}^d$ is the output space of predicted meteorological variables, parametrized by θ, is trained to approximate the mapping from lagged features z_t to the future state y_t.

To avoid temporal leakage, all experiments were conducted using chronological splitting. The dataset was ordered by timestamp before feature generation and model evaluation. The earliest observations were used for training, the subsequent temporal block was used for validation and hyperparameter selection, and the final temporal block was held out exclusively for testing. Specifically, the split was defined as follows: the first 70% of chronologically ordered observations were used for training, the following 15% for validation, and the final 15% for testing. No random shuffling was applied at any stage.

All preprocessing operations that require parameter estimation, including scaling and model fitting, were fitted only on the training block and then applied to validation and test blocks. Hyperparameters were selected using the validation block only, and the final reported performance was computed on the temporally held-out test block.

For robustness analysis, the same chronological protocol was preserved across all degradation scenarios. Synthetic missingness and noise were introduced after defining the temporal blocks, so that information from the validation or test periods could not affect training-time preprocessing. This design ensures that the evaluation reflects a causal one-step-ahead forecasting setting rather than an optimistic randomly shuffled regression task.

$$\theta^* = \arg\min_\theta \frac{1}{N}\sum_{t=1}^{N} \| y_t - f_\theta(z_t) \|^2 \qquad (6)$$

where θ^* are the optimal model parameters, N is the number of training samples, y_t is the ground-truth target, $f_\theta(z_t)$ is the predicted value, and $\|\cdot\|^2$ is the squared Euclidean norm, with time-series-consistent splitting, ensuring that training and validation sets respect chronological order.

The formulation is generic and applies uniformly to all models evaluated in this study, including linear regressors, ensemble methods, and kernel-based predictors. To emulate the operational constraints of fog and/or edge devices, only a short rolling buffer representing a fraction $\alpha = 0.10$ of the most recent samples was used for training:

$$D_{\text{train}} = \{(z_t, y_t) \mid t \in [T(1-\alpha), T]\} \qquad (7)$$

where D_{train} is the training subset, T is the total number of time steps, and α is the retained fraction of the most recent observations reflecting memory limits of fog/edge devices, T denotes the total number of time steps.

This setup reflects realistic memory limitations of embedded IoT gateways, which typically retain only short historical windows due to limited RAM and a lack of persistent storage.

Three forms of data degradation were modelled to replicate common real-world sensor failure modes. Missing-value corruption was introduced via a Bernoulli masking process:

$$\tilde{z}_t = M_t \odot z_t, M_t \sim \text{Bernoulli}(1 - p_{\text{miss}}) \qquad (8)$$

where $\tilde{z}_t$ is the masked feature vector, M_t is the binary mask, p_{miss} is the missing-data probability, and $\odot$ denotes element-wise multiplication simulating signal loss or packet drop, mimicking the effects of intermittent wireless connectivity, packet loss, or power fluctuations. Additive measurement noise was simulated as Gaussian perturbations:

$$\tilde{z}_t = z_t + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma^2 I) \qquad (9)$$

where $\tilde{z}_t$ is the noise-corrupted feature vector, ε_t is Gaussian noise, σ^2 is noise variance, and I is the identity matrix assuming independent noise across all feature dimensions, representing sensor drift, thermal noise, and short-term electrical interference.

The Combined scenario applied both operators simultaneously, reflecting the multifaceted degradation typical of field-deployed meteorological stations.

In summary, the forecasting task involves learning a low-latency, memory-efficient, and degradation-tolerant mapping f_θ capable of operating under noisy measurements, partial observation loss, and limited computational resources, conditions characteristic of real-world fog and/or edge IoT deployments.

 doi:10.20944/preprints202605.0372.v1

To mimic real fog and/or edge constraints, only the most recent 10% of observations were used for training (fast_fraction = 0.10). This scenario reflects practical limitations, such as small memory footprints, limited storage capacity, and low-power processors, which are typical of distributed IoT gateways and microcontrollers. This setup reflects realistic memory constraints of edge devices with limited RAM.

Feature engineering, scenario generation, and model pipelines were implemented in Python using pandas, numpy, scikit-learn, and joblib. These libraries represent standard tools for high-quality time-series ML research.

Experiments were conducted in a Python 3 environment using pandas, NumPy, scikit-learn, and joblib. The evaluation was performed offline on CPU-based hardware. Therefore, the reported results primarily characterize predictive accuracy and robustness under controlled degradation scenarios. They should not be interpreted as direct evidence of deployment feasibility on specific embedded devices. Hardware-level validation, including inference latency, memory footprint, throughput, and energy consumption on representative edge/fog platforms, remains necessary before operational deployment.

The evaluated models include: Ridge Regression, ElasticNet, Lasso, MultiTaskElasticNet, PLSRegression, Support Vector Regression (RBF kernel), Linear SVR, KNN, Random Forest, Extra Trees, Gradient Boosting, and HistGradientBoosting. The selection of these algorithms is motivated by prior research in environmental and agroclimatic monitoring, where regularized linear models and ensemble-based approaches consistently demonstrate high predictive accuracy and robustness under noise and missing data. Earlier studies also show that methods such as Ridge Regression, ElasticNet, Random Forest, Gradient Boosting and HistGradientBoosting achieve stable performance while remaining computationally efficient for deployment on edge and fog devices [10–12]. In agricultural sensing applications, kernel-based and multivariate models have been successfully applied to capture nonlinear relationships and multi-parameter dependencies [16,17]. Considering these findings and the constraints of low-power embedded hardware, this set of algorithms represents a well-justified and practical choice for edge and/or fog-based agroclimatic forecasting.

The ML models evaluated in this study were configured exactly as implemented in the experiment pipeline. Hyperparameter values were selected using a constrained configuration strategy rather than an exhaustive search. The goal was to compare model families under computationally realistic settings suitable for deployment-oriented analysis. Therefore, the selected values reflect commonly used stable configurations in scikit-learn and moderate model sizes that avoid excessive memory and inference costs. This design favors reproducibility and practical comparability, but it does not claim that each model was individually optimized to its maximum possible performance. Consequently, the reported results should be interpreted as a comparison of practical model configurations rather than as a fully exhaustive hyperparameter-optimization study.

Linear and kernel-based models were also included: SVR with an RBF kernel (C = 3.0), LinearSVR (C = 2.0), and PLSRegression with 8 latent components for extracting the multivariate structure. Additionally, ElasticNet and MultiTaskElasticNet were evaluated with l1_ratio = 0.3 to balance L1/L2 regularization. KNN was implemented with 5 neighbors. Models sensitive to feature scaling (Ridge, ElasticNet, Lasso, MultiTaskElasticNet, PLSRegression, SVR, LinearSVR, KNN) were implemented via pipelines with StandardScaler to ensure numerical stability during training.

For each fold, the following standard forecasting metrics were computed:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i | \qquad (10)$$

where n is the number of samples, y_i is the true value, and $\hat{y}_i$ is the predicted value.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (11)$$

where RMSE is the root mean squared error computed from n true–predicted pairs.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \qquad (12)$$

 doi:10.20944/preprints202605.0372.v1

where R^2 is the coefficient of determination, $\bar{y}$ is the true-value mean, and the numerator and denominator represent the model error and total variance respectively.

These metrics reflect absolute error, penalization of large deviations, and the proportion of explained variance.

To evaluate robustness under realistic IoT conditions, the following scenarios were simulated:

1. Baseline: clean data without degradation.
2. Noise: added white Gaussian noise with σ = 0.01–0.15.
3. Missing Data: random masking of 5–30% of values followed by causal forward-fill imputation within each temporal block; initial missing values were handled using statistics estimated from the training block only.
4. Fraction of Data: training on 10–100% of the available history.
5. Combined Distortion: simultaneous application of noise and missing data.

This experimental protocol addresses the core challenges of IoT and fog/edge sensing systems, including intermittent data, sensor drift, packet loss, and constrained computational budgets.

Figure 3 presents an overview of the full research workflow, including data acquisition, preprocessing, feature engineering, scenario simulation, model training, and evaluation. This structured pipeline reflects real fog and/or edge operational constraints and provides reproducible methodology for environmental time-series forecasting.

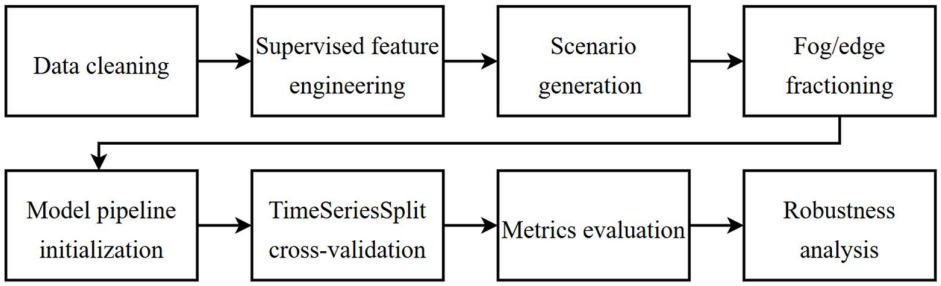


Figure 3. A general research workflow of the experiment.

3. Results

To evaluate the performance and robustness of ML models under realistic fog/edge conditions, we constructed an experimental framework that simulates various types of data degradation commonly encountered in IoT sensor networks. The study includes five experimental scenarios: Baseline, Missing Data, Noise, Fraction of Data, and Combined Distortion. These scenarios represent the most frequent causes of degradation in field-deployed monitoring systems, where sensors may experience packet loss, calibration drift, interference, and power-related inconsistencies.

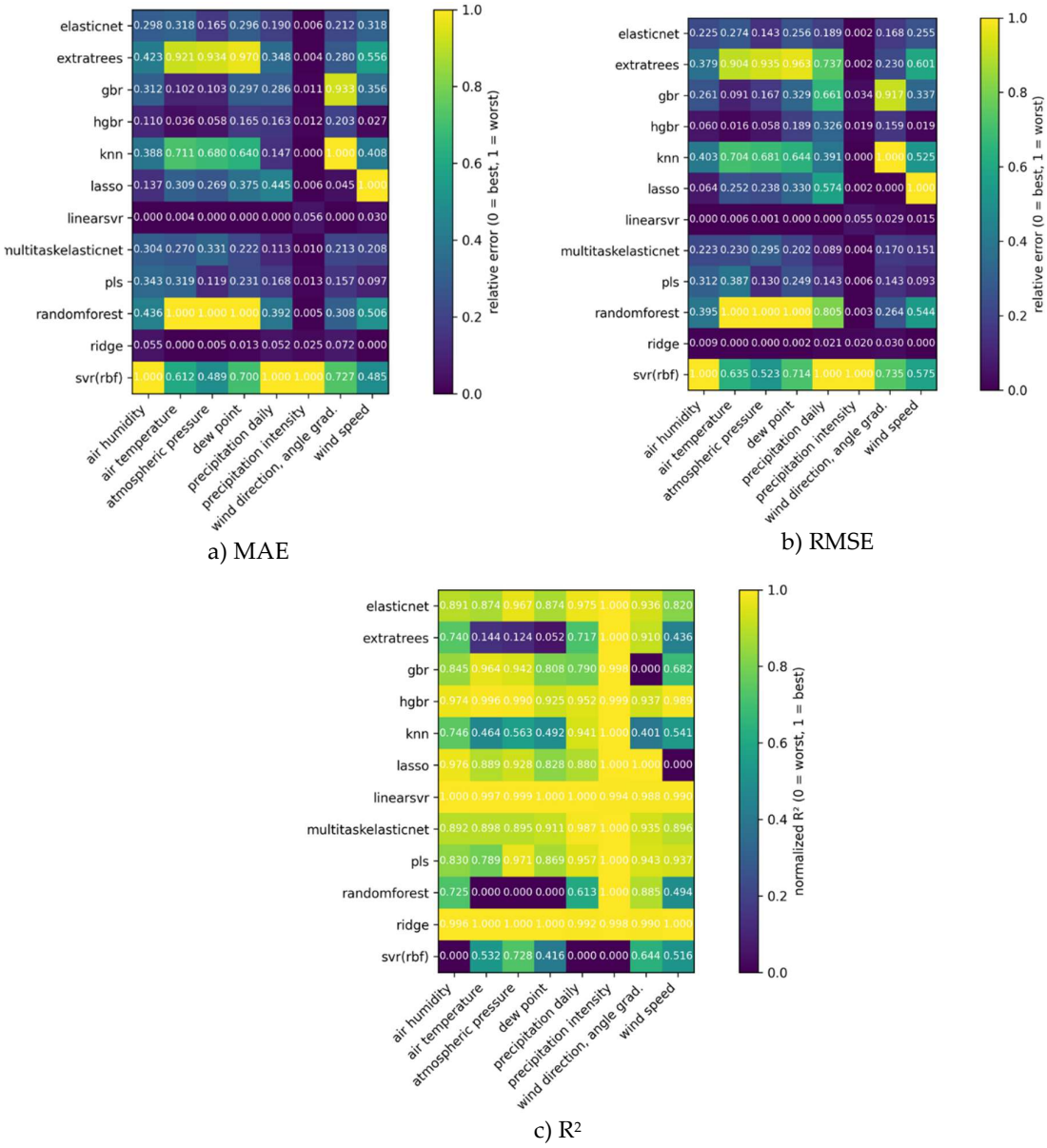
Table 2. Experimental degradation scenarios.

Scenario	Description
Baseline	Clean dataset without any artificial distortions. Serves as the reference for all comparisons.
Missing data	Random removal of 5–30% of sensor readings followed by causal forward-fill imputation within each temporal block; initial missing values were handled using statistics estimated from the training block only. Represents packet loss and intermittent sensor failures.
Noise	Addition of Gaussian noise with σ = 0.01–0.15 to simulate sensor drift and atmospheric interference.

 doi:10.20944/preprints202605.0372.v1

Fraction of data	Restriction of the training window to 10–100% of historical data to emulate edge devices with limited storage.
Combined distortion	Simultaneous application of noise (σ = 0.05–0.1) and missing data (10–20%). Represents the most realistic conditions for IoT field applications.

Because the absolute values of MAE, RMSE, and R^2 differ substantially across meteorological parameters, especially due to the extremely high variance of wind direction, the heatmaps in Figure 4(a–c) present normalized rather than raw metric values. Each heatmap is normalized independently within every target variable so that the results reflect the relative ranking of models rather than the magnitude of the underlying error. For MAE and RMSE, normalization rescales each column to the 0–1 interval, where lower values indicate better performance. The absolute numerical results remain available in the dataset and are used in the quantitative analysis.



a) MAE

b) RMSE

c) R^2

Figure 4. Heatmaps of MAE, RMSE, and R^2 for baseline scenario.

 doi:10.20944/preprints202605.0372.v1

The color scale represents model performance normalized to the $0-1$ range. Axis labels and units: air temperature (°C), air humidity (%), atmospheric pressure (hPa), wind speed (m/s), wind direction (° – angle), precipitation intensity (mm/h), daily precipitation (mm/day), dew point (°C).

Wind direction requires special methodological treatment because it is a circular variable. Values close to 0° and 360° are physically close but numerically distant when evaluated using ordinary regression metrics. Therefore, the poor performance observed for wind direction in the initial degree-based evaluation should not be interpreted as definitive evidence that this variable is intrinsically unpredictable. Rather, it indicates that direct regression on angular degrees may be unsuitable for this target.

In the revised analysis, wind direction should be represented using sine and cosine components, or evaluated using circular error metrics such as mean angular error. Until such circular treatment is implemented, conclusions regarding wind-direction predictability should be considered preliminary and should not be used as a central argument in the robustness analysis.

As shown in the heatmaps above, wind direction exhibits extremely high MAE and RMSE values, as well as strongly negative R^2 values for every evaluated model under the direct degree-based representation. However, these results should be interpreted cautiously because ordinary regression metrics do not account for circular geometry. Therefore, this analysis primarily demonstrates the limitations of direct angular regression rather than proving the intrinsic unpredictability of wind direction.

Figure 5 provides a comparative set of actual–predicted scatterplots for all evaluated models in the Baseline scenario. Across all evaluated algorithms, including linear, ensemble-based, kernel-based, and multivariate methods, the resulting point clouds do not demonstrate any identifiable correlation between true and predicted values. Instead of forming a diagonal pattern that would indicate a learnable relationship, each scatterplot presents a diffuse and amorphous distribution.

The degree-based scatterplots in Figure 5 should therefore be interpreted as a diagnostic visualization of the limitations of direct angular regression rather than as final evidence of wind-direction unpredictability. Since ordinary regression metrics do not account for circular geometry, the wind-direction results are excluded from the main robustness-based model ranking and are discussed separately as a methodological limitation.

This analysis provides an important foundation for interpreting subsequent degradation experiments. Variables whose predictability is affected by representation-specific limitations, such as wind direction under direct degree-based encoding, cannot deteriorate substantially under conditions of noise, missing values, or reduced training history. In contrast, variables with strong temporal coherence, such as temperature and humidity, provide a reliable basis for assessing the robustness and degradation sensitivity of the evaluated models.

 doi:10.20944/preprints202605.0372.v1

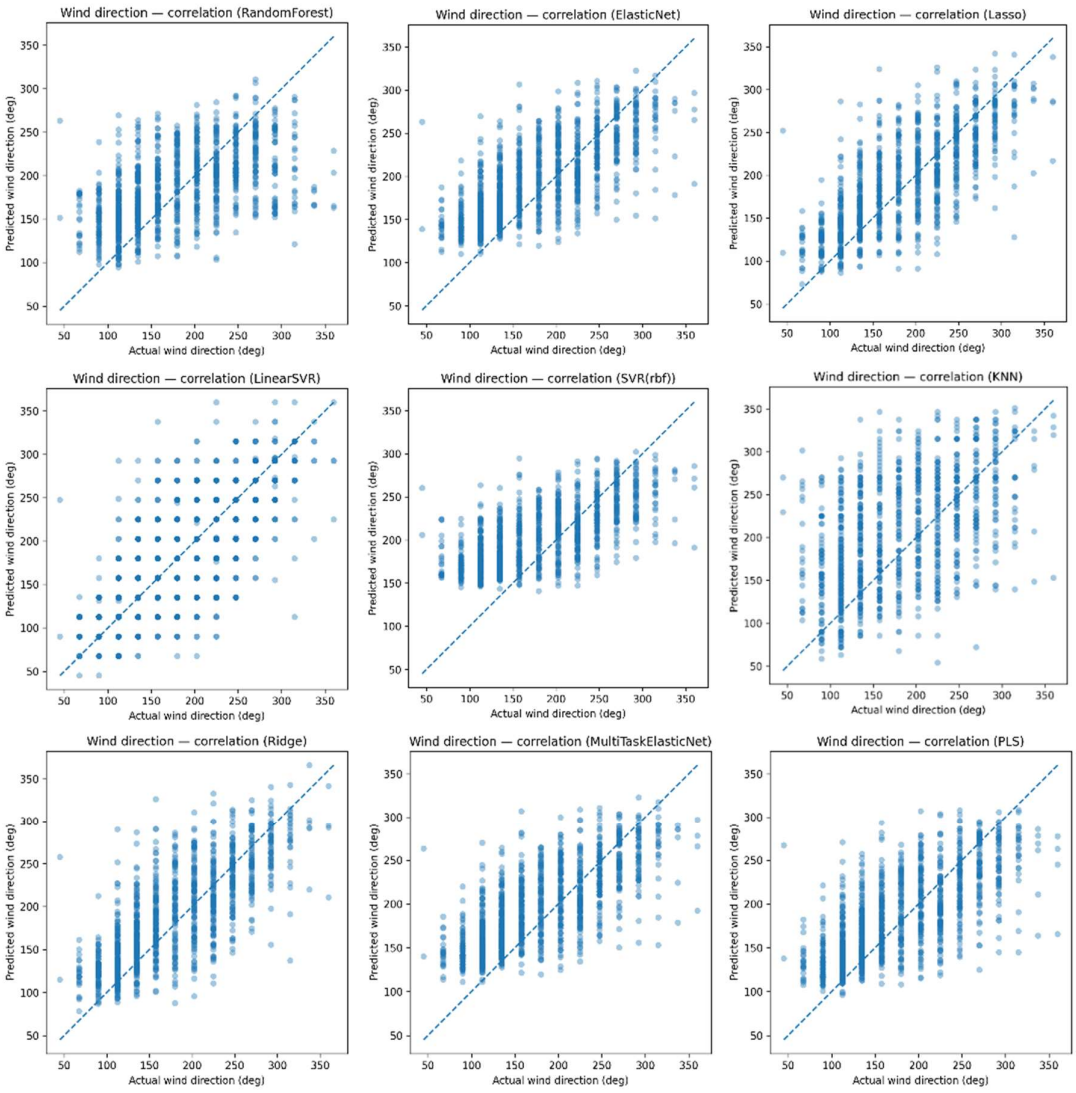


Figure 5. Actual versus predicted wind direction for all evaluated models in the Baseline scenario.

Each panel shows a scatter plot of individual test samples, where the horizontal axis corresponds to the actual wind direction (°) and the vertical axis to the predicted wind direction (°). The blue dots represent single observations, while the dashed diagonal line $y = x$ denotes perfect agreement between prediction and measurement. Points lying close to this line indicate accurate forecasts, whereas large vertical or horizontal deviations reflect higher angular error and poorer model performance.

To obtain a unified comparative picture of how each machine-learning model responds to the full spectrum of data-quality degradation, we constructed three-dimensional surface visualizations based on relative performance metrics. Unlike the earlier heatmaps, which summarize model rankings only under clean Baseline conditions, the 3D surfaces extend the same 0–1 normalization framework across all experimental scenarios, including Missing Data, Noise, Fraction of Available Data, Combined Distortion, and the Baseline reference. This unified scaling enables direct comparison of the surfaces with the heatmaps, while simultaneously visualizing the effect of different degradation regimes in a continuous and interpretable manner. In the surface plots, the horizontal axis represents the model family and the degradation scenario, while the vertical axis encodes the relative performance metric. This structure reveals global robustness patterns that are not easily

 doi:10.20944/preprints202605.0372.v1

visible in two-dimensional figures, for example, monotonic deterioration under increasing noise, sensitivity ridges in fraction-limited learning, or sudden failure under high rates of missing values. Figures 6–8 show the relative-performance landscapes for MAE, RMSE, and R^2, respectively. Although the surfaces represent relative 0–1 normalized values instead of absolute metric values, they clearly highlight how different model classes diverge under adverse conditions. In the MAE and RMSE surfaces, Ridge Regression and HistGradientBoosting consistently occupy regions near the bottom of the surface, indicating low relative forecast error. In contrast, SVR, KNN, and Random Forest form pronounced peaks under strong degradation scenarios, reflecting a rapid loss of predictive stability. In contrast, the R^2 surfaces for Ridge Regression and HistGradientBoosting exhibit elevated regions, indicating high explanatory power and strong predictive consistency, whereas low-lying regions correspond to model failure and loss of variance under severe degradation. The combined view provided by these surfaces allows straightforward identification of degradation-specific vulnerabilities and the generalization capabilities of each algorithm across heterogeneous IoT conditions. To facilitate the interpretation of the three-dimensional surfaces in Figures 6–8, the following notation is used for the horizontal axis representing degradation scenarios: baseline denotes clean reference data; missing x% corresponds to removal of x% of samples; noise σ refers to additive Gaussian perturbation with standard deviation σ; fraction y% indicates training on only y% of the available history; and combined denotes simultaneous missing values and injected noise.

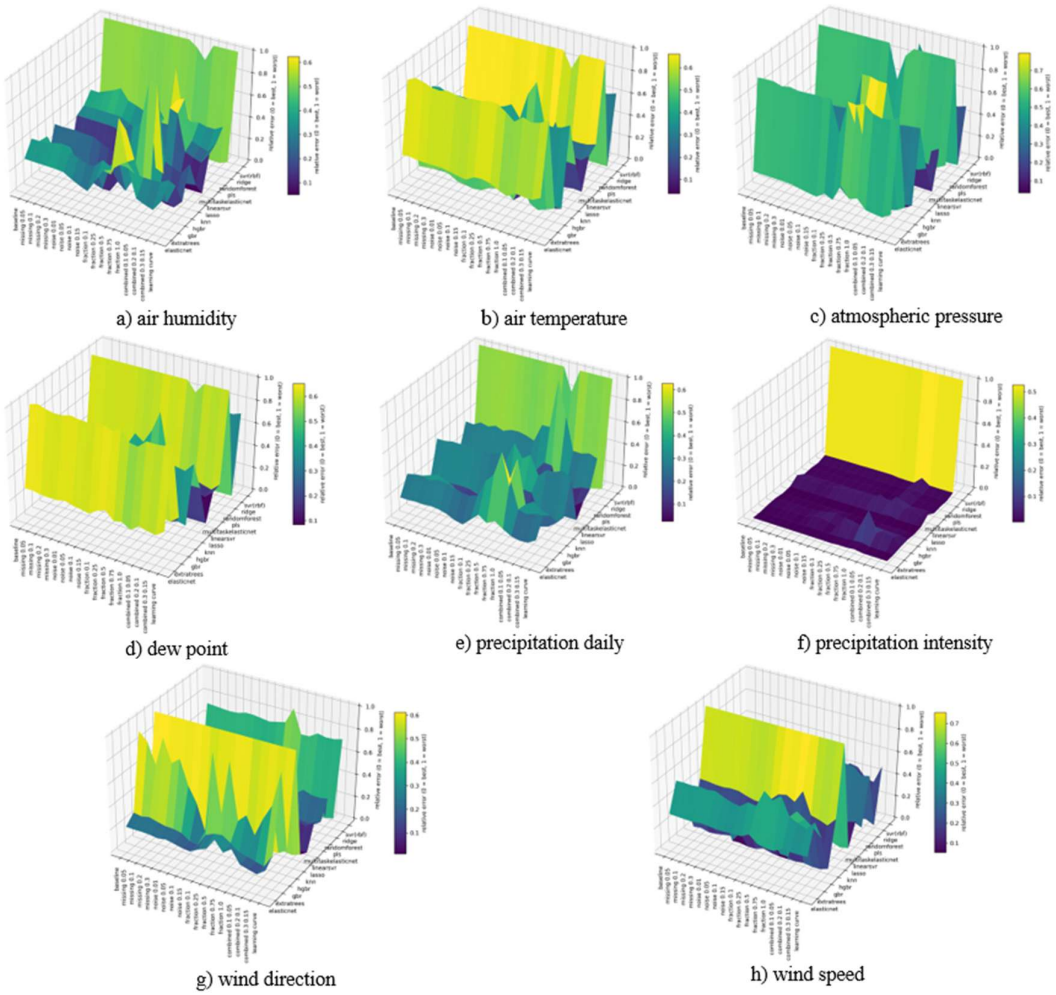


Figure 6. Relative MAE across all models and degradation scenarios

 doi:10.20944/preprints202605.0372.v1

In these surfaces, the horizontal axis represents the ordered sequence of degradation scenarios, including the Baseline reference and progressively more adverse Missing, Noise, Fraction-of-data, and Combined conditions. The vertical axis enumerates all evaluated machine-learning models, while the height and color of the surface encode the relative MAE on a normalized 0–1 scale, where lower values indicate better performance. Darker regions correspond to models that maintain small absolute errors under the given scenario, whereas elevated bright regions highlight configurations that deteriorate sharply when data become incomplete, noisy, or limited. This visualization, therefore, reveals the stability landscape of each algorithm across increasingly challenging degradation regimes.

Lower values indicate smaller absolute forecast errors and therefore better short-term predictive accuracy. Models positioned near the lower regions of the surface demonstrate stable performance across degradation scenarios, whereas peaks correspond to scenarios where models experience a strong increase in MAE due to noise, missing data, or limited training history.

.

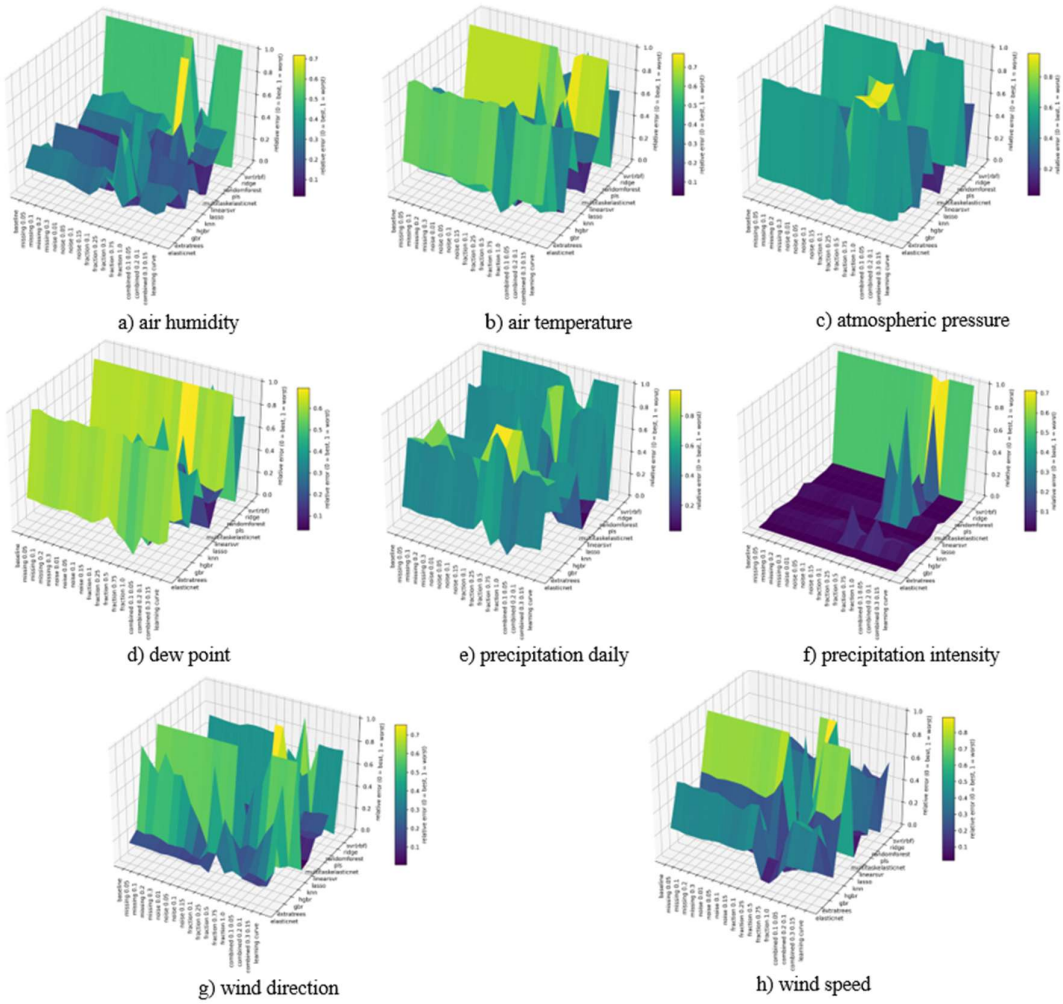


Figure 7. Relative RMSE across all models and degradation scenarios.

Here, the horizontal axis shows the degradation scenarios ordered from the Baseline to the most challenging Combined conditions, while the vertical axis lists the machine-learning models. The surface height and color reflect the normalized RMSE, expressed on a 0–1 scale in which lower values correspond to stronger robustness against large deviations. Dark continuous regions indicate models that maintain a low RMSE even when the data are corrupted or reduced, whereas steep peaks indicate configurations that

 doi:10.20944/preprints202605.0372.v1

become highly sensitive to noise, missingness, or limited training fractions. The surfaces thus provide an interpretable view of error amplification patterns across models and scenarios.

The relative RMSE highlights sensitivity to large deviations in the forecast. Lower values correspond to models that effectively suppress extreme errors, while higher ridges indicate instability under specific degradation regimes. Because RMSE penalizes occasional large fluctuations more strongly than MAE, the surface reveals which models are particularly vulnerable to outliers in noisy or incomplete data.

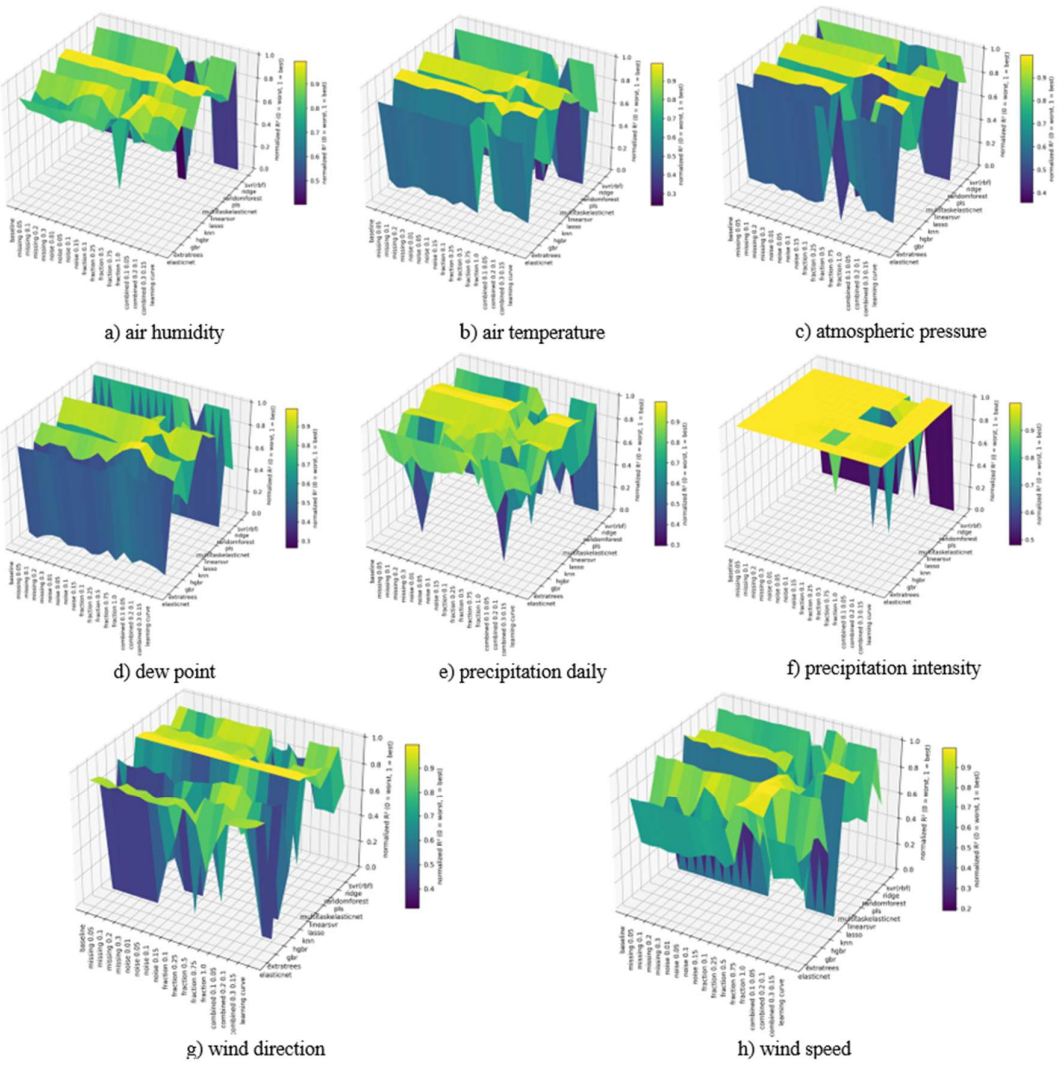


Figure 8. R^2 across all models and degradation scenarios.

In these plots, the horizontal axis denotes the full spectrum of degradation scenarios, and the vertical axis contains the set of evaluated models. The surface height and color encode the R^2 values on the 0–1 scale, where higher values reflect stronger predictive consistency and higher original R^2, whereas lower values correspond to degraded explanatory power. Bright elevated regions reveal models that preserve variance under adverse conditions, while dark depressions mark regimes in which predictive structure collapses.

Higher R^2 values indicate better variance preservation and stronger explanatory power of the model. Regions with low R^2 (close to 0) correspond to scenarios where a model fails to capture meaningful temporal structure, often performing worse than a naive mean predictor. The surface clearly illustrates differences in generalization capability under increasing levels of noise, missing data, and restricted training history.

 doi:10.20944/preprints202605.0372.v1

In the Baseline condition, the strongest performance for temperature forecasting was obtained by regularized linear models, particularly Ridge Regression, which achieved an MAE of 0.318 and an R^2 of approximately 0.98. This suggests that, for the one-step-ahead horizon and the engineered lag-based feature space used in this study, short-term temperature dynamics can be captured effectively by a linear model. This interpretation should be restricted to the present dataset, forecast horizon, and feature construction strategy.

Under 5–15% missing values, Ridge Regression remained stable, with MAE increasing by less than 10%. HistGradientBoosting surpassed Ridge only at higher missing rates (20–30%). Conversely, models such as SVR, KNN, and Random Forest exhibited a sharp decline in performance, with R^2 falling to zero or even becoming negative at 30% missing values, indicating a severe loss of predictive capability.

When Gaussian noise with $\sigma = 0.05–0.1$ was injected into the dataset, both Ridge Regression and Hist Gradient Boosting demonstrated robust behavior, maintaining R^2 in the range of 0.95–0.97. In contrast, SVR and Random Forest showed strong degradation, with negative R^2 values indicating worse performance than the naive mean prediction.

Ridge Regression achieved an R^2 value greater than 0.9 even when trained on only 50% of the available data, confirming its ability to generalize under a reduced training history—an essential requirement for fog/edge devices with limited memory. RandomForest, ExtraTrees, and SVR failed to generalize when training data fell below 50%, reflecting their higher variance and sensitivity to limited datasets.

The Combined scenario, which included simultaneous noise and missing values, provided the strongest separation between the evaluated models. Ridge remained effective for temperature and humidity, but for more volatile parameters (e.g., wind or precipitation), HistGradientBoosting outperformed it. Under the combined degradation scenario with $\sigma = 0.1$ noise and 20% missing data, HistGradientBoosting achieved an R^2 value greater than 0.85 in the evaluated test configuration, whereas several alternative models showed substantially lower or even negative R^2 values. This result indicates that HistGradientBoosting is a strong candidate for severely degraded conditions in the present benchmark. However, broader claims about its general superiority require validation across additional temporal folds, seasons, and external datasets.

Table 3. The best-performing models across experimental scenarios.

Scenario	Best model	Reason for superiority
Baseline	Ridge Regression	Highest accuracy (MAE = 0.318, $R^2 \approx 0.98$) on clean data.
Missing data	Ridge (≤15%), HGBR (>20%)	Ridge stable under small missing rates; HGBR excels under heavy missingness.
Noise	Ridge / HGBR	Best robustness to Gaussian noise $\sigma = 0.05–0.1$; minimal drop in R^2.
Fraction of data	Ridge Regression	Maintains $R^2 > 0.9$ with only 50% training data; low sensitivity to data reduction.
Combined distortion	HistGradientBoosting	Best-performing model in the evaluated test configuration; maintained $R^2 > 0.85$ under combined noise + missing data.

The experimental results obtained in this study not only enable the comparison of the robustness of different machine-learning models under realistic degradation conditions but also form the basis for a practical decision-making workflow suitable for edge and/or fog deployments. Since IoT meteorological stations often operate under constrained communication, storage, and energy

 doi:10.20944/preprints202605.0372.v1

budgets, the forecasting pipeline must adapt dynamically to changes in data quality and environmental conditions.

Based on the observed scenario-dependent performance of Ridge Regression and HistGradientBoosting, we propose an adaptive prediction workflow that selects the appropriate forecasting model depending on the detected degradation regime. In this paradigm, the system continuously collects sensor measurements at the edge node, performs lightweight preprocessing (denoising, imputation, rolling statistics), and then evaluates key diagnostic indicators such as: percentage of missing samples; estimated noise intensity (based on z-score residuals or variance drift); available portion of historical buffer; resource constraints.

These indicators are mapped to one of the five scenario families evaluated in this study. If the system detects clean or near-clean conditions, a lightweight linear model (Ridge Regression) is selected to ensure low computational cost and high accuracy. Under moderate degradation (missing values or noise), Ridge remains the preferable option. In contrast, under heavy degradation or simultaneous distortions, the system switches to HistGradientBoosting, which has been shown to maintain predictive power even when R^2 drops close to zero for other models.

The proposed workflow should be interpreted as a conceptual deployment architecture derived from the observed benchmark results. It illustrates how an edge or fog node could select between a lightweight model and a more robust fallback model based on data-quality indicators. However, the present study does not yet provide an end-to-end implementation of the switching controller, threshold optimization, switching-cost analysis, or hardware-level runtime validation. Therefore, the workflow is presented as a practical design implication and a direction for future deployment-oriented research rather than as a fully validated adaptive system.

Such a mechanism is essential for distributed agricultural, environmental, and industrial IoT systems, where global connectivity cannot be guaranteed, and models must operate reliably on-device.

To translate these findings into a deployable fog/edge-oriented solution, we outline an adaptive forecasting workflow suitable for real-time operation on resource-constrained devices. The conceptual architecture of the proposed system is presented in Figure 9. In this workflow, raw meteorological measurements are first acquired at the edge node and passed through a lightweight preprocessing module that is responsible for denoising, handling missing values, and computing rolling statistical features using a short sliding buffer. This ensures that the subsequent modelling steps receive a stable and temporally consistent input stream.

The preprocessed data are then routed to a data-quality assessment module, which estimates key diagnostic indicators including the proportion of missing samples, the effective noise level derived from residual-based statistics, the size of the available local historical buffer, and the current computational state of the device (CPU load, memory availability, energy budget). These indicators are mapped to one of the degradation regimes examined in this study: baseline, missing data, noise, fraction of data, or combined distortion via a rule-based classification component.

Depending on the detected mode, the model-selection unit activates the forecasting model that is most suitable for the current data quality conditions. Under clean or mildly degraded inputs, the system selects Ridge Regression, which provides low-latency inference and high predictive accuracy. Under severe degradation such as substantial missingness, strong Gaussian noise, or their combination, the system switches to HistGradientBoosting, which demonstrated superior robustness across all high-distortion scenarios. The selected model performs the one-step-ahead prediction, which is subsequently used either locally (e.g., for control decisions in agricultural systems) or transmitted to cloud services for logging, visualization, or offline retraining.

From an implementation standpoint, the workflow can be deployed as a modular edge-resident software pipeline in which preprocessing, diagnostics, regime classification, and model inference are executed at fixed sampling intervals. The models themselves can be pre-trained offline and deployed in serialized form (e.g., joblib, ONNX, or TensorFlow Lite), enabling efficient runtime inference on devices with limited computational resources. This architecture ensures that the forecasting system

 doi:10.20944/preprints202605.0372.v1

remains fully operational under intermittent connectivity conditions, adhering to the strict latency, memory, and power constraints characteristic of fog/edge environments.

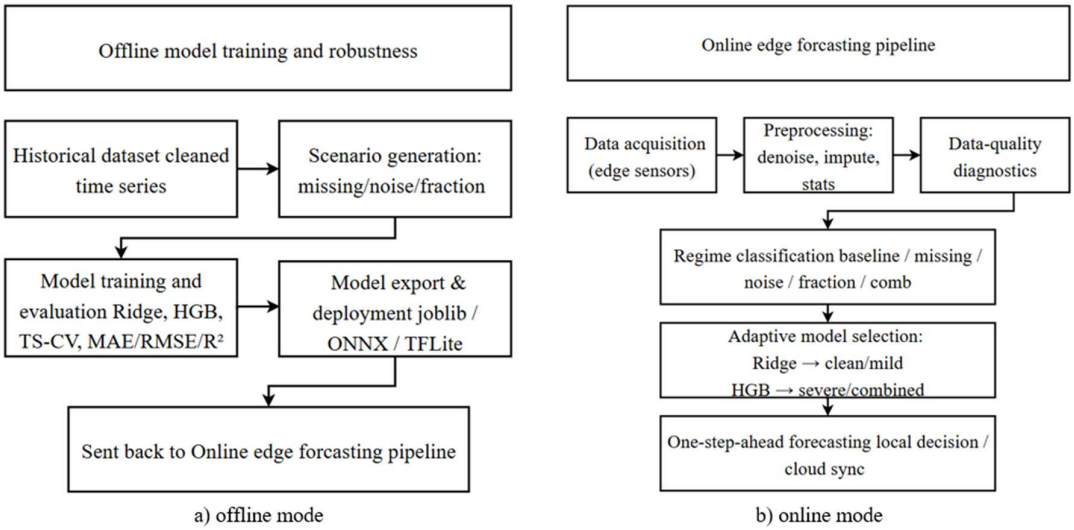


Figure 9. Adaptive edge/fog forecasting workflow.

4. Discussion and Prospects for Further Research

The experimental evaluation demonstrates that different ML models exhibit distinct strengths depending on the type and severity of data degradation, a crucial characteristic for real-world fog/edge deployments, where sensors frequently produce incomplete or noisy data. Among the evaluated models, Ridge Regression [13] consistently achieved the highest accuracy on clean or mildly corrupted data. Its stability under low to moderate levels of degradation (≤15% missing values, moderate Gaussian noise) suggests that the short-term meteorological dependencies captured by the lag-based feature space exhibit a predominantly linear structure. This observation aligns with prior studies, which have shown that environmental variables with high temporal autocorrelation are effectively modeled using regularized linear approaches.

In contrast, HistGradientBoosting (HGBR) [19,20] provided the strongest robustness against combined distortions, simultaneous noise, and missing data – conditions that closely emulate real IoT field deployments. Its tolerance to structural irregularities arises from the boosting ensemble mechanism, which captures complex nonlinear interactions while maintaining relative insensitivity to local perturbations. This explains why HGBR was the only model capable of maintaining $R^2 > 0.85$ under the most challenging Combined scenario, where other models collapsed to near-zero or negative predictive performance. These findings corroborate earlier observations that boosting-based models outperform classical tree ensembles in the presence of sensor-level noise and degraded data streams [11,21].

Models such as SVR, KNN, Random Forest, and Extra Trees showed substantial variability across scenarios. Their strong performance on clean data did not generalize to noisy or incomplete data. Such behavior is undesirable for fog/edge systems, where retraining opportunities are limited, and data irregularities are common. The pronounced instability of these models underlines the importance of robust feature engineering and regularization when working with short and distorted time series.

The fraction-of-data experiment revealed another critical dimension: the ability to learn from restricted training histories. Ridge Regression maintained performance even with only 50% of the training data, validating its suitability for edge devices with limited memory or storage resources. Boosting-based models degraded more gradually, while high-variance approaches (Random Forest,

 doi:10.20944/preprints202605.0372.v1

Extra Trees) failed to generalize with smaller datasets. These results highlight a practical trade-off for deployment: models with controlled complexity and strong regularization are preferable when training data are limited.

Although the obtained results provide clear insights into the behavior of ML models under various degradation scenarios, several practical limitations should be acknowledged. First, the experiments rely on a single real-world dataset collected from a single agricultural location, which may limit the generalizability of the findings to regions with different climatic profiles or sensor infrastructures. Second, the study focuses exclusively on short-term (one-step-ahead) forecasting, leaving multi-horizon prediction unexplored. Third, the degradation scenarios were simulated in a controlled manner, whereas real fog and/or edge deployments often involve complex failure patterns, non-Gaussian noise, sensor drift over time, and asynchronous data streams. Finally, the evaluation was performed offline; therefore, computational latency, online retraining feasibility, and energy consumption on actual edge hardware require further validation before large-scale operational deployment.

Overall, the results suggest adopting a dual-model operational strategy in fog and/or edge environments: Ridge Regression for normal or mildly corrupted data streams, and HGBR for scenarios with significant degradation. Such adaptive switching requires only lightweight monitoring of data quality (share of missing values, noise levels, degradation of online validation metrics) and aligns well with the operational constraints of low-power IoT systems.

Future research may extend these findings in several promising directions. One important avenue is the development of adaptive or self-correcting forecasting pipelines capable of detecting data degradation in real time and dynamically switching between models, such as Ridge Regression and HGBR, based on predefined criteria. Another direction involves integrating advanced feature extraction techniques (wavelets, seasonal decomposition, learned embeddings) to improve robustness under non-stationary or highly volatile environmental conditions. Additionally, evaluating model performance on multiple weather stations, diverse climates, and heterogeneous IoT networks would help validate scalability across broader deployment contexts. Finally, implementing lightweight online learning or incremental training strategies could further enhance the practicality of these models for long-term fog and/or edge operation with continuously drifting sensor data.

5. Conclusions

This study provides a comprehensive evaluation of ML models for short-term forecasting of environmental time series under fog and/or edge constraints, incorporating multiple types of realistic sensor data degradation. Using real meteorological datasets and controlled degradation scenarios, we compared the performance of Ridge Regression, Elastic Net, Lasso, Multi-Task Elastic Net, PLS Regression, Support Vector Regression, Linear SVR, KNN, Random Forest, Extra Trees, Gradient Boosting, and HistGradientBoosting across multiple degradation scenarios.

The key findings are as follows:

1. Ridge Regression demonstrates the most reliable performance on clean and mildly degraded data, confirming its suitability as a baseline model for edge deployments.
2. HistGradientBoosting is the most robust model in adverse conditions, maintaining $R^2 > 0.85$ under combined noise and missing data. Its resilience to structural irregularities makes it particularly suitable for real-world IoT deployments with unstable sensor behavior.
3. High-variance models (RandomForest, ExtraTrees, SVR, KNN) suffer significant performance deterioration under noise and missing values, often producing negative R^2 scores. These models are not recommended for fog/edge forecasting unless complemented by advanced preprocessing or adaptive training.
4. Reduced training history impacts models differently: Ridge Regression and HGBR retain predictive power even with 50% of the data, while other models fail to generalize. This makes them practical for edge devices with limited memory.

 doi:10.20944/preprints202605.0372.v1

A practical and efficient deployment approach is a two-tier forecasting system, where Ridge Regression is used as a fast, low-cost baseline, and HGBR serves as a robust fallback model activated when data degradation is detected.

Author Contributions: Conceptualization, I.L.; methodology, O.Z.; software, O.Z.; validation, G.D., I.L., O.V.; formal analysis, D.M.; investigation, O.Z., I.L., G.D.; data curation, O.V., D.M.; writing—original draft preparation, O.Z.; writing—review and editing, I.L., G.D.; visualization, O.Z., D.M.; supervision, I.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was carried out as part of the scientific project 'Software and algorithmic solutions for agricultural information and analytical systems based on artificial intelligence in variable agroclimatic conditions' funded by the Ministry of Education and Science of Ukraine at the expense of the state budget (0126U001122).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mao, Y., You, C., Zhang, J., Huang, K. & Letaief, K. B. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322–2358 (2017). DOI: 10.1109/COMST.2017.2745201
2. Laroui, M., Nour, B., Moungla, H., Cherif, M. A., Afifi, H. & Guizani, M. Edge and fog computing for IoT: A survey on current research activities & future directions. *Computer Communications* 180, 210–231 (2021). DOI: 10.1016/j.comcom.2021.09.003
3. Kalyani, Y. & Collier, R. A systematic survey on the role of cloud, fog, and edge computing combination in smart agriculture.
   Sensors 21(17), 5922 (2021). DOI: 10.3390/s21175922
4. Adhikari, D., Jiang, W., Zhan, J., He, Z., Rawat, D. B., Aickelin, U. & Khorshidi, H. A. A comprehensive survey on imputation of missing data in Internet of Things. *ACM Computing Surveys* 55(7), Article 133, 1–38 (2022). DOI: 10.1145/3533381
5. Alajlan, N. N. & Ibrahim, D. M. TinyML: Enabling of inference deep learning models on ultra-low-power IoT edge devices for AI applications. *Micromachines* 13(6), 851 (2022). DOI: 10.3390/mi13060851
6. Shi, W., Cao, J., Zhang, Q., Li, Y. & Xu, L. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646 (2016). DOI: 10.1109/JIOT.2016.2579198
7. Voyant, C., Notton, G., Kalogirou, S., Nivet, M. L., Paoli, C., Motte, F. & Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105, 569–582 (2017). DOI: 10.1016/j.renene.2016.12.095
8. Szostek, K., Mazur, D., Drałus, G. & Kusznier, J. Analysis of the effectiveness of ARIMA, SARIMA, and SVR models in time series forecasting: A case study of wind farm energy production. *Energies* 17(19), 4803 (2024). DOI: 10.3390/en17194803
9. Li, G. & Yang, N. A hybrid SARIMA–LSTM model for air temperature forecasting. *Advanced Theory and Simulations* 6(2), 2200502 (2023). DOI: 10.1002/adts.202200502
10. Wang, X., Han, Y., Leung, V. C. M., Chen, X., Zhou, Z. & Jiao, L. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials* **22**(2), 869–904 (2020). DOI: 10.1109/COMST.2020.2970550
11. Breiman, L. Random forests. *Machine Learning* 45, 5–32 (2001). DOI: 10.1023/A:1010933404324
12. Cortes, C. & Vapnik, V. Support-vector networks. *Machine Learning* 20, 273–297 (1995). DOI: doi.org/10.1007/BF00994018

 doi:10.20944/preprints202605.0372.v1

13. Hoerl, A. E. & Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(1), 55–67 (1970). URL: https://homepages.math.uic.edu/~lreyzin/papers/ridge.pdf
14. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5), 1189–1232 (2001). DOI: 10.1214/aos/1013203451
15. Lim, B. & Zohren, S. Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379(2194), 20200209 (2021). DOI: 10.1098/rsta.2020.0209
16. Laktionov, I., Vizniuk, A. & Diachenko, G. Researching ML Algorithms for Predicting FHB in Corn: A Case Study in Dnipro Region of Ukraine. In: *Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds) Artificial Intelligence and Soft Computing. ICAISC 2024. Lecture Notes in Computer Science* 15164, 167–183 (2025). DOI: 10.1007/978-3-031-84353-2_15
17. Diachenko, G., Laktionov, I., Vovna, O., Aleksieiev, O. & Moroz, D. Computer Model of an IoT Decision-Making Network for Detecting the Probability of Crop Diseases. *IoT* 6 (1), 1–23 (2025). DOI: 10.3390/iot6010008
18. Diachenko, G., Laktionov, I., Vinyukov, O. & Likhushyna H. A Decision Support System for Wheat Powdery Mildew Risk Prediction Using Weather Monitoring, Machine Learning and Explainable Artificial Intelligence. *Computers and Electronics in Agriculture* 230, 1–24 (2025). DOI: 10.1016/j.compag.2025.109905
19. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 3146–3154 (2017). URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
20. Ross, A., Pan, W., Celi, L. & Doshi-Velez, F. Ensembles of locally independent prediction models. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(4), 5527–5536 (2020). DOI: 10.1609/aaai.v34i04.6004
21. Fynn, M., Nordholm, S. & Rong, Y. Coherence function and adaptive noise cancellation performance of an acoustic sensor system for use in detecting coronary artery disease. *Sensors* 22(17), 6591 (2022). DOI: 10.3390/s22176591