Weather Forecasting Using a PCA-Based Optimized Random Forest Classification Technique

Chandan kumar Mahto[1], Dr. Renu Bagoria[2]

[1]M.Tech. CSE Scholar, Department of Computer Science, Jagan Nath University, Jaipur
[2] Professor-CSE, Department of Computer Science, Jagan Nath University, Jaipur

Abstract:
Accurate weather prediction is essential for agriculture, aviation, transportation and disaster management, yet conventional statistical and physics-based forecasting models struggle to capture the non-linear and highly correlated relationships that exist among meteorological variables such as temperature, humidity, wind speed and atmospheric pressure. This paper reports an experimental study of a classification-based weather forecasting system built around a Principal Component Analysis–optimized Random Forest (PCA-ORF) model. The raw meteorological records are first normalized and cleaned, after which Principal Component Analysis (PCA) is used to compress the correlated input attributes into a smaller set of uncorrelated components. These components are then supplied to an optimized Random Forest classifier, whose hyperparameters are tuned through cross-validation, to predict the precipitation category. The model is trained and evaluated on the publicly available "Weather in Szeged 2006–2016" dataset comprising 96,453 hourly observations across twelve attributes. PCA-ORF model reaches 99.67% accuracy (a 0.33% error rate), 100% precision, 88.84% recall and an F1 value of 94.09%, outperforming a previously reported ensemble-based forecasting approach that achieved 95.00% accuracy and a 5.00% error rate on a comparable task. The results confirm that combining PCA-based dimensionality reduction with an optimized Random Forest classifier yields a forecasting model that is markedly more accurate and reliable than the conventional ensemble baseline.

Keywords: Weather Forecasting; Machine Learning; Principal Component Analysis; Random Forest; PCA-ORF; Classification; Feature Reduction

I. INTRODUCTION

Weather forecasting refers to the scientific task of estimating future atmospheric conditions for a given location and time horizon, and it underpins decision-making in domains ranging from agriculture and aviation to disaster preparedness and energy management. Forecasts have traditionally been generated through physics-based models remain the backbone of operational meteorology, they are computationally expensive and can struggle to represent the highly non-linear and interdependent behaviour of variables such as temperature, humidity, wind speed etc.

Over the past decade, the growing availability of historical meteorological records, combined with advances in computing power, has encouraged a shift toward data-driven forecasting using machine learning (ML). Rather than encoding atmospheric physics explicitly, ML models learn statistical regularities directly from historical observations, making them attractive for short-term, localized forecasting tasks such as predicting whether it will rain, snow, or remain dry. Tree-based ensemble methods, and Random Forest (RF) in

particular, are especially well suited to this setting because they can model non-linear feature interactions, handle a mixture of numeric and categorical attributes, and resist overfitting better than a single decision tree by aggregating the votes of many trees trained on different data subsets.

A practical difficulty when applying Random Forest directly to raw meteorological data is that weather attributes differ greatly in scale and are often mutually correlated — for instance, pressure and wind bearing span numerically larger ranges than humidity or temperature, and several attributes vary together in predictable seasonal patterns. Feeding such heterogeneous, correlated features directly into a classifier can bias tree-splitting decisions toward attributes with larger numeric ranges rather than those with genuine predictive value.

Building on this observation, this paper presents an experimental evaluation of a PCA-Based Optimized Random Forest (PCA-ORF) classification pipeline for weather-category prediction. The specific contributions of this work are: (i) a pre-processing and dimensionality-reduction pipeline that normalizes raw meteorological attributes and compresses them using PCA before classification; (ii) an optimized Random Forest classifier whose parameters are tuned through cross-validation to improve generalization and reduce overfitting; and (iii) a quantitative comparison of the proposed model against a previously reported ensemble-based forecasting method on standard classification metrics.

II. RELATED WORK

A substantial body of work has applied classical supervised-learning algorithms — Linear Regression, Decision Trees, Support Vector Machines and Random Forest — to weather-related classification and regression tasks. Kumar et al. [1] compared several such algorithms for general weather prediction using temperature, humidity, wind speed and pressure as inputs, evaluating models with accuracy, F1-score and Area-Under-Curve. Chen et al. [2] proposed a lightweight forecasting approach for settings where only limited historical data is available, comparing missing-value handling strategies alongside Random Forest and Support Vector Machine regressors. Mahajan et al. [3] applied a classification approach to rainfall prediction with an 80:20 train–test split and reported roughly 85% accuracy, while Barua et al. [5] used weather-driven modelling to size hybrid wind–solar renewable-energy systems for a coastal site in Bangladesh. Collectively, these studies establish that classical ML models can extract useful predictive signal from meteorological attributes, though reported accuracies for pure classification tasks on comparable data generally remain below the high-90% range.

A second stream of research applies deep-learning architectures to weather and related time-series prediction problems, exploiting their ability to model long-range temporal or spatial dependencies. Goel et al. [4] used deep convolutional networks to detect severe weather events in climate simulation archives with 98.5% accuracy, while Li et al. [6] showed that LSTM networks outperform linear and support-vector baselines for hourly air-temperature forecasting. Hostetter and Angryk [7] used Generative Adversarial Networks to synthesize rare solar-flare samples for training CNN-based space-weather classifiers, and Sleeman et al. [8] applied CNNs to LiDAR measurements for atmospheric boundary-layer detection. Jiang et al. [9] fused radar, optical-satellite and SAR imagery through a multisource CNN to detect thunderstorms and gale-force winds. These deep architectures generally require larger, richer datasets and greater computational resources than tree-based ensembles, which motivates continued interest in lighter-weight ensemble methods for tabular meteorological data.

Most closely related to the present study are works that pair Random Forest with a dimensionality-reduction step. Irmanda et al. [10] combined Random Forest with Chi-Square feature selection for rainfall-type classification, reporting close to 88% accuracy but noting overfitting on minority rainfall classes. Zhou et al. [11] embedded PCA within a hybrid VMD-PCA-RF pipeline to correct numerical-weather-prediction wind-speed forecasts, obtaining forecast accuracy above 85% with stable error levels across seasons. Wang et al. [12] and Drisya et al. [13] used Random Forest respectively to bias-correct NWP wind/pressure fields and to forecast wind speed from short historical windows, while Barrera-Animas et al. [14] benchmarked several contemporary ML algorithms for rainfall time-series forecasting without settling on one universally superior model. Outside meteorology proper, Chen et al. [15], Hoque et al. [16] and Pham et al. [17] each combined PCA with Random Forest or related ensemble models to improve climate-driven crop-yield prediction, consistently finding that PCA-reduced inputs generalize better than raw, correlated climate features.

Taken together, the reviewed literature shows that Random Forest and PCA are individually well established for weather- and climate-related prediction, and that combining them has already proven beneficial in adjacent problems such as wind-speed correction [11] and crop-yield estimation [15], [17]. However, relatively few studies treat PCA and Random Forest as a single, jointly optimized pipeline specifically for weather-category classification, and fewer still report a full set of classification-oriented metrics (precision, recall and F1-score, in addition to accuracy) alongside a direct, like-for-like comparison against a conventional ensemble baseline. The present work addresses this gap by evaluating a PCA-ORF pipeline against an ensemble baseline under an identical evaluation protocol.

III. MATERIALS AND METHODS

The proposed research methodology follows a sequential pipeline consisting of dataset acquisition, pre-processing, train–test partitioning, PCA-based dimensionality reduction, Random Forest classification with hyperparameter optimization, and performance evaluation, as shown in Fig. 1.

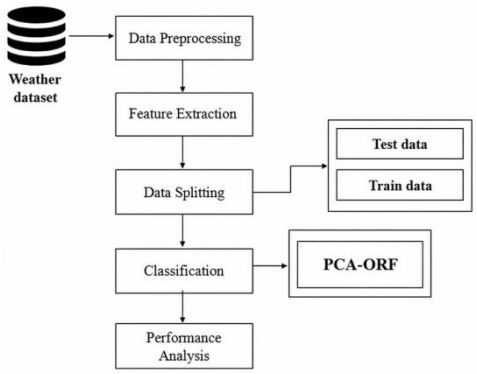

Fig. 1. Block diagram of the proposed PCA-ORF research methodology.

A. Dataset Description

All experiments draw on the "Weather in Szeged 2006-2016" dataset, a public collection available through Kaggle [18]. It holds 96,453 hourly weather readings gathered over ten years, described by the twelve fields listed in Table I. The label being predicted is precipitation type — rain, snow or none — treated here as a three-way classification target.

TABLE I. SUMMARY OF DATASET ATTRIBUTES

S. No.	Attribute	Description
1	Formatted Date	Timestamp of the recorded observation
2	Summary	Short textual weather condition (e.g., Partly Cloudy)
3	Precip Type	Type of precipitation (rain / snow / none) — target class
4	Temperature (°C)	Recorded air temperature
5	Apparent Temperature (°C)	"Feels-like" temperature
6	Humidity	Relative humidity (0–1 scale)
7	Wind Speed (km/h)	Wind speed in kilometres per hour
8	Wind Bearing (degrees)	Wind direction in degrees
9	Visibility (km)	Visibility distance in kilometres
10	Loud Cover	Cloud-cover indicator
11	Pressure (millibars)	Atmospheric pressure
12	Daily Summary	Long-form textual summary of the day's weather

B. Data Pre-processing

Prior to model development, the raw dataset is cleaned and prepared through the following steps: (i) numerical attributes such as pressure, wind bearing and wind speed — which naturally span much larger ranges than humidity or temperature — are standardized to a common scale so that no single feature dominates model training purely because of its magnitude; (ii) categorical attributes such as the weather Summary field are converted into a numeric representation suitable for the Random Forest algorithm; (iii) missing or inconsistent entries are handled to prevent them from biasing the learned model; and (iv) the class distribution of the target variable is inspected for imbalance, since precipitation categories such as snow occur far less frequently in the dataset than the dominant no-precipitation class.

C. Reducing Dimensions with PCA

Once the fields are on a common scale, Principal Component Analysis is applied to the resulting matrix. PCA identifies a set of orthogonal directions (principal components) along which the variance of the data is maximized, and re-expresses each observation as a linear combination of these components rather than the original, correlated attributes. Retaining only the components that capture the bulk of the total variance allows the twelve raw meteorological attributes to be represented more compactly while discarding redundant or noisy variation. This step is particularly relevant here because several raw attributes — for example temperature and apparent temperature, or wind speed and wind bearing — are correlated with one another, and training a classifier directly on such correlated inputs can dilute the effective contribution of each feature during tree splitting.

D. Train–Test Partitioning

The pre-processed dataset is partitioned into training and testing subsets using a 70:30 split, with the split performed on a shuffled version of the data so that the training and test partitions are drawn randomly rather

than from a contiguous time block. The training subset is used exclusively to fit the PCA transformation and the Random Forest classifier, while the held-out test subset is used only at evaluation time.

E. PCA-Based Optimized Random Forest (PCA-ORF) Classifier

Classification itself rests on a Random Forest: a bundle of decision trees, each grown on a bootstrap sample of the training rows and a random slice of the available fields at every split. Once trained, every tree in the forest casts a vote for an incoming record, and the class receiving the most votes becomes the prediction. Because no two trees see quite the same data or the same fields, the forest as a whole is far less likely to overfit than any single tree would be, and it copes reasonably well with noisy or missing values.

The "optimized" part of PCA-ORF comes from tuning the forest's settings — how many trees, how deep they grow, how many fields are considered at each split — through k-fold cross-validation rather than guesswork. The training data is repeatedly divided into folds, the model trained on some and checked against the rest, and whichever setting combination holds up best across folds is carried forward into the final model. Putting this tuned forest together with the PCA-reduced inputs from Section III-C gives the PCA-ORF classifier used for all reported results.

F. Performance Evaluation Metrics

Performance is evaluated using the confusion matrix and four metrics derived from it — Accuracy, Precision, Recall and F1-score — together with the complementary Error Rate. Let TP, TN, FP and FN denote the true-positive, true-negative, false-positive and false-negative counts respectively; the metrics are defined as:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$Precision = TP / (TP + FP) \quad (2)$$

$$Recall = TP / (TP + FN) \quad (3)$$

$$F1\text{-}score = 2 \cdot TP / (2 \cdot TP + FP + FN) \quad (4)$$

$$Error\ Rate = 100 - Accuracy \quad (5)$$

Accuracy summarizes overall correctness but can be misleading under class imbalance; precision tells us how trustworthy a positive prediction is, which matters when false alarms carry a cost; recall tells us how many real occurrences of a class actually get caught, which matters for something as consequential as a storm warning; and F1-score, being their harmonic mean, gives a single number that reflects both at once — useful given the class imbalance in the precipitation labels used here

IV. EXPERIMENTAL SETUP

The proposed PCA-ORF pipeline was implemented in Python 3.7 using the Spyder IDE, with the scikit-learn, NumPy, pandas, Matplotlib and Seaborn libraries used for data handling, model development and result visualization [19]. The system configuration used for all experiments is summarized in Table II.

TABLE II. EXPERIMENTAL SYSTEM CONFIGURATION

Sr. No.	Parameter	Value
1	Operating System	Windows 10
2	RAM	8 GB
3	Processor	Intel Core i3
4	Programming Language	Python
5	IDE	Spyder

Sr. No.	Parameter	Value
6	Python Version	3.7

V. RESULTS AND DISCUSSION

A. Overall Classification Performance

Table III lists how the PCA-ORF classifier performed on the 30% of records held back for testing.

TABLE III. OVERALL PERFORMANCE OF THE PROPOSED PCA-ORF MODEL

Sr. No.	Metric	Value (%)
1	Accuracy	99.67
2	Error Rate	0.33
3	Precision	100.00
4	Recall	88.84
5	F1-Score	94.09

The model attains an overall classification accuracy of 99.67%, corresponding to an error rate of only 0.33%, indicating that the large majority of test samples are assigned to their correct precipitation category. A precision of 100% shows that whenever the model predicts a particular weather class, that prediction is essentially always correct, leaving little room for false positives. The recall of 88.84% is noticeably lower than the precision, indicating that the model misses a small but non-trivial proportion of the true instances of at least one class; as discussed below, this pattern is consistent with the class imbalance present in the dataset, in which one precipitation category is far less frequent than the others and is therefore intrinsically harder to recall completely. The F1-score of 94.09%, being the balance point between precision and recall, still points to strong overall performance despite that gap.

B. Confusion Matrix Analysis

Fig. 2 shows the confusion matrix for the test set as a colour-coded heat map. Class 0 has 87 true cases, the model correctly classifies 82 and misclassifies the remaining 5 as class 1, a recall of roughly 94% for this minority class. Class 1, which dominates the dataset with 34,216 actual instances, is predicted correctly in 34,097 cases, with only 119 instances incorrectly assigned to class 0 — a recall of about 99.7%. Class 2, comprising 4,279 instances, is classified with perfect accuracy, with every instance correctly recovered. The near-total absence of confusion between class 2 and the other two classes indicates that the meteorological conditions defining this category are well separated in the PCA-reduced feature space, whereas the comparatively higher confusion between classes 0 and 1 suggests that these two categories share more overlapping feature characteristics — a pattern that is consistent with the lower overall recall reported in Table III.

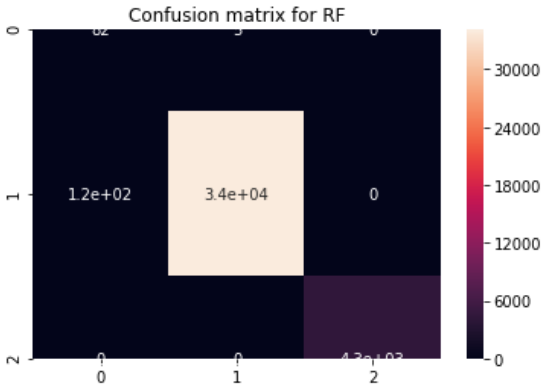

Fig. 2. Confusion-matrix heat map for the proposed PCA-ORF model

C. ROC Curve Analysis

Fig. 3 plots the ROC) curve relating the True-Positive Rate to the False-Positive Rate. Because this curve is derived from the discrete predicted class labels rather than a continuous probability score, it appears as a small number of straight line segments rather than a smoothly varying curve. The curve passes through an operating point at approximately (0.87, 0.89) on the False-Positive/True-Positive axes before reaching the top-right corner of the plot. The position of this operating point well above the diagonal reference line — which would represent a purely random classifier — confirms that the model discriminates between the weather classes far better than chance, consistent with the high accuracy and precision already reported in Table III.

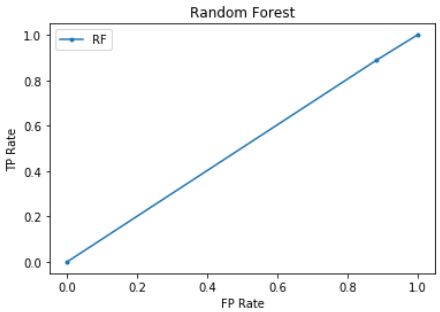

Fig. 3. ROC curve for the proposed Random Forest–based classifier

D. Comparative Analysis

Table IV benchmarks the proposed PCA-ORF model against a previously reported ensemble-based forecasting method [1] evaluated on a comparable weather-classification task.

TABLE IV. PERFORMANCE COMPARISON WITH PREVIOUS WORK

Sr. No.	Parameter	Previous Work [1]	Proposed PCA-ORF
1	Method	Ensemble	PCA-ORF
2	Accuracy (%)	95.00	99.67
3	Error Rate (%)	5.00	0.33

While the earlier ensemble-based method reports an accuracy of 95% and a corresponding error rate of 5%, the proposed PCA-ORF model raises the accuracy to 99.67% and lowers the error rate to 0.33% on a comparable weather-classification task — a relative reduction in error of over 93%, meaning the proposed model misclassifies roughly one in every three hundred test instances against roughly one in every twenty for the previous approach. Fig. 4 visualizes this comparison as bar charts of accuracy and error rate. The

improvement can reasonably be attributed to the PCA-based dimensionality-reduction step incorporated ahead of the Random Forest classifier, which removes redundant and correlated information among the twelve raw meteorological attributes before the ensemble-learning stage, allowing the individual decision trees to split on more informative, decorrelated components rather than on noisy or overlapping raw features.

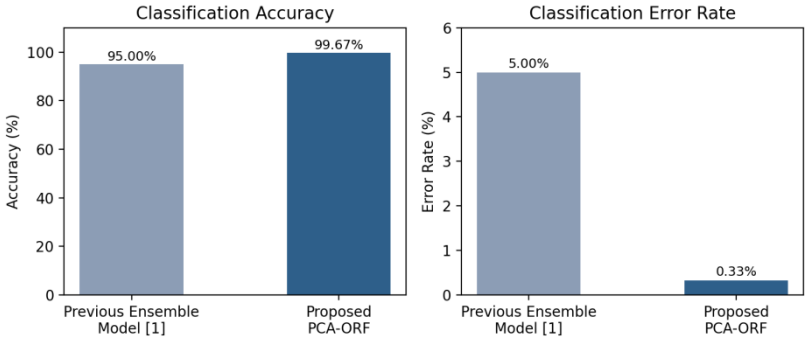

Fig. 4. Comparison of classification accuracy and error rate between the previous ensemble method [1] and the proposed PCA-ORF model

E. Discussion

Overall, the experimental results indicate that pairing PCA-based dimensionality reduction with an optimized Random Forest classifier yields substantial gains over a conventional ensemble baseline for weather-category classification, both in overall accuracy and in the balance between precision and recall captured by the F1-score. The gap between the near-perfect precision and the comparatively lower recall highlights that further improvement is most likely to come from addressing class imbalance in the training data — for instance through oversampling of minority precipitation classes or cost-sensitive learning — rather than from the classification algorithm itself. These findings are consistent with related studies that pair PCA with Random Forest for wind-speed correction and crop-yield estimation [11], [15], [17], reinforcing the broader conclusion that PCA-based feature reduction is a valuable pre-processing step whenever Random Forest is applied to correlated, multi-scale meteorological data.

VI. CONCLUSION AND FUTURE SCOPE

This paper set out to test a PCA-Optimized Random Forest pipeline for weather-category classification, and the results back the idea up: scaling and cleaning the raw fields, cutting them down with PCA, and classifying the result with a cross-validation- tuned Random Forest ensemble produced 99.67% accuracy and a 0.33% error rate on held-out Szeged weather data, alongside 100% precision, 88.84% recall and a 94.09% F1-score. Against a previously published ensemble baseline at 95.00% accuracy and 5.00% error, that is over a 93% cut in classification error — reasonable evidence that folding dimensionality reduction into an optimized ensemble beats treating the two as separate, unconnected steps.

A few directions could take this further: bringing in extra data sources such as satellite or radar imagery alongside the tabular fields used here; trying other ensemble methods (Gradient Boosting, stacking) in place of or alongside Random Forest; testing deep architectures such as LSTM or CNN for capturing weather patterns that unfold over time and space; balancing the classes to lift recall on the rarer precipitation types without giving up precision; building a streaming version of the pipeline suited to edge or IoT weather sensors; and reporting predictions as calibrated probabilities rather than flat class labels, which would suit risk-sensitive uses such as disaster planning.

REFERENCES:

[1] N. Kumar et al., "Weather Forecasting Using Machine Learning Techniques," in Proc. Int. Conf. on Computing Methodologies and Communication.

[2] B. Chen et al., "An Innovative Method for Weather Forecasting Using Limited Data," in Proc. Int. Conf. on Data Engineering.

[3] D. Mahajan et al., "Precipitation Forecasting Using a Machine Learning Classification Approach," in Proc. Int. Conf. on Computing and Communication.

[4] S. Goel et al., "Deep Learning for Identifying Severe Weather Events in Climate Datasets," in Proc. IEEE Conf. on Big Data.

[5] P. Barua et al., "Weather-Dependent Modelling of Hybrid Wind–Solar Renewable Energy Systems for Coastal Bangladesh," in Proc. Int. Conf. on Renewable Energy.

[6] C. Li, M. Zhao, Y. Liu and F. Xu, "Air Temperature Forecasting Using Traditional and Deep Learning Algorithms," in Proc. 7th Int. Conf. on Information Science and Control Engineering (ICISCE), Changsha, China, 2020, pp. 189–194, doi: 10.1109/ICISCE50968.2020.00049.

[7] M. Hostetter and R. A. Angryk, "First Steps Toward Synthetic Sample Generation for Machine Learning Based Flare Forecasting," in Proc. IEEE Int. Conf. on Big Data, Atlanta, USA, 2020, pp. 4208–4217, doi: 10.1109/BigData50022.2020.9377986.

[8] J. Sleeman, M. Halem, Z. Yang, V. Caicedo, B. Demoz and R. Delgado, "A Deep Machine Learning Approach for Lidar Based Boundary Layer Height Detection," in Proc. IGARSS IEEE Int. Geoscience and Remote Sensing Symp., Waikoloa, USA, 2020, pp. 3676–3679, doi: 10.1109/IGARSS39084.2020.9324191.

[9] Y. Jiang, J. Yao and Z. Qian, "A Method of Forecasting Thunderstorms and Gale Weather Based on Multisource Convolution Neural Network," IEEE Access, vol. 7, pp. 107695–107698, 2019, doi: 10.1109/ACCESS.2019.2932027.

[10] H. N. Irmanda, E. Ermatita, M. K. bin Awang and M. Adrezo, "Enhancing Weather Prediction Models through the Application of Random Forest Method and Chi-Square Feature Selection," JOIV: Int. J. on Informatics Visualization, vol. 8, no. 3-2, pp. 1506–1514, Nov. 2024, doi: 10.62527/joiv.8.3-2.2356.

[11] S. Zhou, C. Y. Gao, Z. Duan, X. Xi and Y. Li, "A Robust Error Correction Method for Numerical Weather Prediction Wind Speed Based on Bayesian Optimization, Variational Mode Decomposition, Principal Component Analysis, and Random Forest: VMD-PCA-RF," Geoscientific Model Development, vol. 16, pp. 6247–6259, 2023, doi: 10.5194/gmd-16-6247-2023.

[12] A. Wang, L. Xu, Y. Li, J. Xing, X. Chen, K. Liu, Y. Liang and Z. Zhou, "Random-Forest Based Adjusting Method for Wind Forecast of WRF Model," Computers & Geosciences, vol. 155, p. 104842, 2021, doi: 10.1016/j.cageo.2021.104842.

[13] G. V. Drisya, P. Valsaraj, K. Asokan and K. Satheesh Kumar, "Wind Speed Forecast Using Random Forest Learning Method," Int. J. on Computer Science and Engineering, 2022, doi: 10.48550/arXiv.2203.14909.

[14] A. Y. Barrera-Animas, L. O. Oyedele, M. Bilal, T. D. Akinosho, J. M. Davila Delgado and L. A. Akanbi, "Rainfall Prediction: A Comparative Analysis of Modern Machine Learning Algorithms for Time-Series Forecasting," Machine Learning with Applications, vol. 7, p. 100204, 2022, doi: 10.1016/j.mlwa.2021.100204.

[15] M. Chen, N. Guilpart and D. Makowski, "Comparison of Methods to Aggregate Climate Data to Predict Crop Yield: An Application to Soybean," Environmental Research Letters, vol. 19, no. 5, p. 054049, 2024, doi: 10.1088/1748-9326/ad42b5.

[16] M. J. Hoque, M. S. Islam, J. Uddin, M. A. Samad, B. Sainz De Abajo, D. L. Ramirez Vargas and I. Ashraf, "Incorporating Meteorological Data and Pesticide Information to Forecast Crop Yields Using Machine Learning," IEEE Access, vol. 12, pp. 47768–47786, 2024, doi: 10.1109/ACCESS.2024.3383309.

[17] H. T. Pham, J. Awange, M. Kuhn, B. Van Nguyen and L. K. Bui, "Enhancing Crop Yield Prediction Utilizing Machine Learning on Satellite-Based Vegetation Health Indices," Sensors, vol. 22, no. 3, p. 719, 2022, doi: 10.3390/s22030719.

[18] budincsevity, "Weather in Szeged 2006–2016," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/budincsevity/szeged-weather

[19] Spyder IDE. [Online]. Available: https://www.spyder-ide.org/