A computerized method for predicting the risk of powdery mildew in wheat based on software analysis of soil and climatic monitoring data*

Grygorii Diachenko[1, *,†], Ivan Laktionov[1,†], Dmytro Moroz[1,†], Maryna Derzhevetska[2,†], Sergii Semenov[1,†]

[1] *Dnipro University of Technology, av. Dmytra Yavornytskoho 19, UA49005, Dnipro, Ukraine*
[2]*Technical University "Metinvest Polytechnic", Southern Highway 80, UA69008, Zaporizhzhia, Ukraine*

Abstract

Today, smart agriculture is one of the core technologies for sustainable development and increasing the efficiency of open-field crop production enterprises of various sizes and forms of ownership in the face of changing climate conditions. The development and implementation of computerized methods and intelligent software and hardware solutions for transforming large volumes of agroclimatic data distributed in time and space is a relevant and important field for improving the efficiency of information technologies for agrotechnical applications. In this article, the scientific and applied problem of creating and validating a computerized method for predicting the probability of occurrence of crop diseases at the pre-symptomatic stage, which forms the basis of software and hardware components for processing data from agromonitoring systems based on fog architecture, has been solved. The main results of the research are: reduction of the number of informative features to five based on the Harris Hawk Optimizer algorithm, proving the effectiveness of Bagged Trees and Medium Neural Network algorithms in the classification of Powdery Mildew in Wheat, synthesis and testing of a computer model in Simulink that implements a full cycle of transformation of agroclimatic monitoring data in predicting the Risk of Powdery Mildew in Wheat. In addition, prospective directions for further research to improve the efficiency of information technologies for predicting the probability of crop diseases are substantiated in the article.

Keywords

Classification, soil and climatic parameters, computerized method, prediction, Powdery Mildew Blumeria Graminis, feature selection, machine learning

1. Introduction

To date, the principles of digitalization and intellectualization of technological processes are one of the global trends in improving the efficiency of production processes at enterprises of various profiles, scales and forms of ownership. Agriculture is one of the strategic sectors of the national economies of many countries, and therefore requires constant search and implementation of scientifically substantiated approaches to the sustainable development of agricultural practices. Smart farming is a key concept relevant to running production processes in agricultural enterprises, particularly open-field crop production. This concept, in turn, is made possible by introducing technologies such as: Internet of Things, machine learning and artificial intelligence, remote sensing, drones and robotics. This approach allows achieving a significant socio-economic, environmental and technological effect, which consists in: efficient use of material, land and labor and time

AdvAIT-2024: 1st International Workshop on Advanced Applied Information Technologies, December 5, 2024, Khmelnytskyi, Ukraine - Zilina, Slovakia
* Corresponding author.
† These authors contributed equally.
diachenko.g@nmu.one (G. Diachenko); laktionov.i.s@nmu.one (I. Laktionov); dmitriy@moroz.cc (D. Moroz); Maryna.Derzhevetska@mipolytech.education (M. Derzhevetska); semenov.s.y@nmu.one (S. Semenov)
0000-0001-9105-1951 (G. Diachenko); 0000-0001-7857-6382 (I. Laktionov); 0000-0003-2577-3352 (D. Moroz); 0000-0002-9952-4992 (M. Derzhevetska); 0000-0002-1244-9687 (S. Semenov)

resources; increasing the resistance of field crops to changing climatic conditions; increasing yields and minimizing negative environmental impact [1, 2].

Based on a priori analysis of current statistics on agricultural practices at the global level, it has been established that cereals (wheat, rice, corn, barley, and others) are the most popular crops in terms of cultivated areas and specific yields [3, 4]. Over the past decade, wheat has been the leader among cereals at the national level in terms of cultivated areas: from 5.28 million hectares to 7.1 million hectares. It is also necessary to emphasize that with an increase in cultivated areas, there is no proportional increase in the yield of cereals [5]. This phenomenon is rooted in the fact that during the full cycle of cultivation, grain crops are subject to destabilizing effects of physical (changing agroclimatic conditions) and biological (pests and diseases) factors, which negatively affect the integral stress resistance and productivity and, as a result, crop yields.

Therefore, an actual scientific and practical task is to develop and implement methods, models, and software and hardware for predicting the occurrence of crop diseases at the pre-symptomatic stage in real time, which will allow timely planning and implementation of agrotechnical measures to increase the stress resistance of crops and preserve the harvest in changing agroclimatic conditions.

In the present-day world practice, there is a significant number of high-quality research and developments of information and computer technologies for agrotechnical purposes to detect the parameters and characteristics of the processes of occurrence and progression of crop diseases based on various data collection technologies [6], in particular: obtaining and analyzing graphic images from satellites [7, 8] and UAVs [9, 10], as well as collecting and processing data from ground-based sensor networks [11, 12].

One of the main tasks, which is the focus of many relevant studies in developing intelligent information technologies for predictive monitoring of crops, is the precise and reliable analysis of observation results. To date, machine learning methods have gained considerable popularity in solving problems of intellectual analysis of large amounts of data when creating intelligent information technologies for various applied fields [13, 14]. In agriculture, such technologies are used for intelligent processing of agroclimatic data distributed in time and space [15, 16], as these approaches allow aggregating, analyzing, and interpreting large amounts of measurement data with subsequent automatic support for making management decisions to optimize agrotechnical procedures.

The perspectives of integrating sensor networks for agroclimatic monitoring and machine learning methods have been proven by the authors of scientific studies on: the introduction of precision farming systems [17], analysis of promising practices for managing agrotechnical processes and resources [18], accounting for the impact of changing climatic conditions on crop cultivation regimes [19], and others.

Thus, the results of the analysis of the current state of scientific and applied research prove the potential and effectiveness of creating and implementing information technologies for predictive monitoring of crop diseases based on online measurements of soil and climatic parameters with their subsequent processing by software based on machine learning algorithms. Consequently, the current research task is to develop computer components for complex intelligent processing of agromonitoring measurement data that implements a full cycle of data transformation (primary statistical processing, selection of informative features, and predictive analytics) within an integrated hardware and software architecture, taking into account the specifics of detected diseases and agroclimatic growing conditions for specific types and periods of grain crops.

Therefore, the aim of the article is to further develop information technologies for agrotechnical purposes to predict the risk of occurrence of grain crop diseases (Powdery Mildew Blumeria graminis in wheat) at the pre-symptomatic stage through the development and research of computerized method of intelligent analysis of measured data of agroclimatic monitoring utilizing machine learning models. The object of research is information processes of software analysis of agroclimatic data distributed in time and space. The subject of research is methods and computer models of complex predictive processing of agroclimatic monitoring data.

Accordingly, researching the development and validation of computerized methods and software components of predictive transformation and analysis of agroclimatic measurement data during the creation of information technologies for agrotechnical purposes to increase the stress resistance of grain crops is an actual scientific and practical task.

2. Materials and methods

Two primary software environments were used in this research: Python 3.10.12 and MATLAB R2024a. Preliminary data analysis was performed in Python using libraries such as Pandas and NumPy. MATLAB was used to train the classifier model with the possibility of further generating code for microcontrollers. For this purpose, the Simulink and Statistics and Machine Learning toolboxes were used.

In Figure 1, a generalized structure of the research in this article is illustrated.

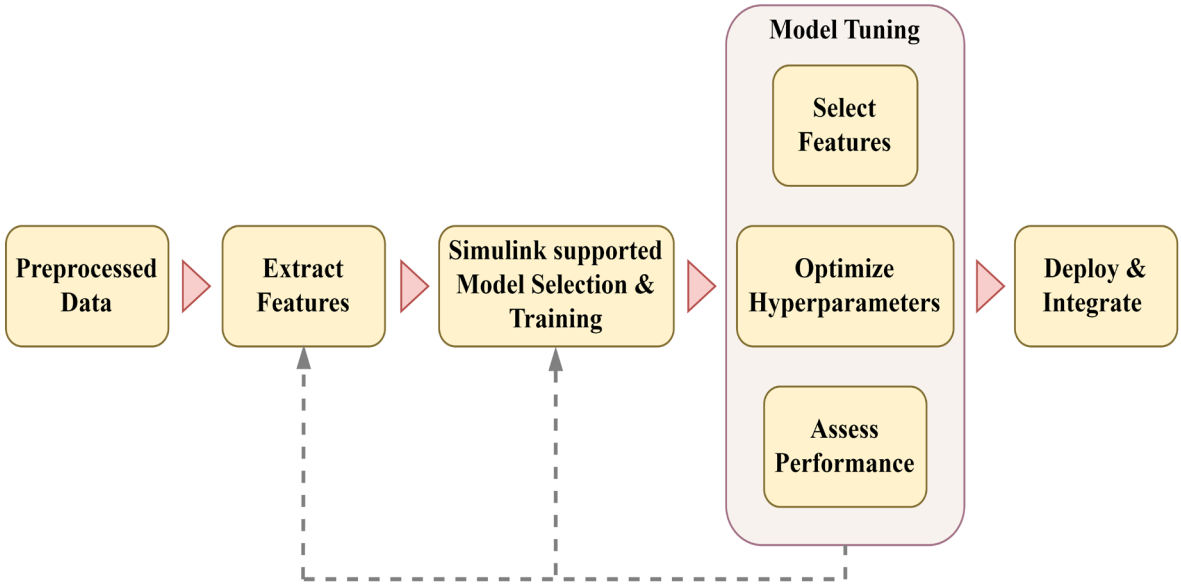


Figure 1: Generalized structure of the research.

The data for the research was obtained using professional Metos weather stations from Pessl Instruments via the FieldClimate IoT platform, access to which was provided by Metos Ukraine LLC. The experimental data reflects the results of monitoring soil and climatic parameters collected in two agroclimatic zones of Ukraine from September 2022 to September 2023:

- northern steppe of Ukraine: a zone characterized by an arid and very warm climate. The hydrothermal coefficient varies from 0.7 to 1.0, and the annual sum of temperatures ranges from 2900 °C to 3300 °C;
- forest-steppe of Ukraine: insufficiently humid and warm zone with a hydrothermal coefficient of 1.0 to 1.3 and an annual sum of temperatures from 2500 °C to 2900 °C.

The data sample from the northern steppe zone (Dnipro region) consisted of 8656 records with a sampling interval of 1 hour and 14 attributes. Similarly, the data sample from the forest-steppe zone (Cherkasy region) consisted of 8687 records with the same sampling interval and number of attributes. Both samples have the probability of occurrence of Powdery Mildew Blumeria graminis as the target function for further analysis and modeling. A description of all attributes included in the two data sets is given in Table 1.

Table 1
Soil and climatic attributes present in the dataset

SI. No	Attribute	Units	Datatype	Description
1	DT	-	Continuous	Date and time
2	AT	°C	Continuous	Air temperature
3	DP	°C	Continuous	Dew point
4	SR	Wt/m^2	Continuous	Solar radiation
5	VPD	kPa	Continuous	Vapor pressure-deficit
6	RH	%	Continuous	Relative humidity of the air
7	PR	mm	Continuous	Precipitation within one hour
8	LW	min	Discrete	Leaf wetness. If the leaves were wet during the last hour (60), otherwise (0)
9	WS	m/s	Continuous	Wind speed
10	WG	m/s	Continuous	Wind direction
11	WD	m/s	Continuous	Wind gust
12	ST	°C	Continuous	Soil temperature
13	ET	mm	Continuous	Evapotranspiration
14	PMBG	%	Continuous	Risk of Powdery Mildew Blumeria graminis disease in range from 0 to 100

During the preliminary data analysis, all rows containing missing values were removed to avoid distortion of the results and to ensure the correctness of the modeling. There were two such rows and given that 'PMBG' is calculated once a day, their removal does not affect the value of the target function. In addition, for the initial analysis of the data in Python, the describe() function from the Pandas library was used to obtain statistical information about the numerical characteristics of the data, such as mean, standard deviation, minimum and maximum values, and quartiles. These indicators allowed assessing data distribution and identifying possible anomalies and general trends in the dataset. The statistical indicators obtained as a result of using describe() for the combined sample from the two regions are shown in Table 2.

Table 2
Descriptive statistics of the data collected (combined data of two regions)

	count	mean	std	min	25%	50%	75%	max
AT		10.6	9.7	-10.9	2.7	9.8	18.2	37
DP		5.3	7.7	-16.5	-0.3	5.5	11.5	25
SR		135.2	215.7	0	0	3	192	1059
VPD		0.5	0.6	0	0.1	0.2	0.7	5
RH		72.6	17.2	19	61	76	87	100
PR		0.1	0.3	0	0	0	0	15.2
LW	17343	8.6	21	0	0	0	0	60
WS		3.5	1.8	0	2.1	3.2	4.6	13.4
WG		6.1	2.7	0.3	4.1	5.8	7.8	19.1
WD		180.2	107.3	1	83	180	278	360
ST		11.9	8.9	-6.5	3.9	11	19.1	35.7
ET		0.1	0.1	0	0	0	0.1	1.3
PMBG		12.2	19.2	0	0	0	20	70

The average 'AT' value of 10.6 °C indicates a moderate climate in the region. The range of values from 10.9°C to 37°C indicates the presence of both cold and very warm periods. The 'AT' values are centered around 9.8°C (median), with the bottom quartile (25%) falling within 2.7°C and the top quartile (75%) falling within 18.2°C, indicating a significant temperature variation. The average 'RH'

value is 72.6%, indicating generally humid conditions. The humidity varies widely, with a minimum of 19% and a maximum of 100%, with most values concentrated between 61% and 87%. A mean 'PR' value of 0.1 mm indicates low precipitation. In fact, most of the records show no precipitation, as the median and 25th quartile are 0. The maximum value of 15.2 mm indicates significant but rare precipitation. The average 'LW' value is 8.6 minutes per hour, indicating predominantly dry conditions. 75% of the 'LW' values are 0, which means that the leaves remain dry most of the time. 75% of 'ET' values below 0.1 mm indicate generally low evapotranspiration. The average 'PMBG' value is 12.2%, indicating a low average risk of developing the disease. The maximum 'PMBG' is 70%, which indicates a significant risk in certain periods. At the same time, the median of 0% indicates that a significant part of the sample had no or low risk of the disease, but the 75th quartile at 20% shows that there are periods with an increased risk of developing the disease.

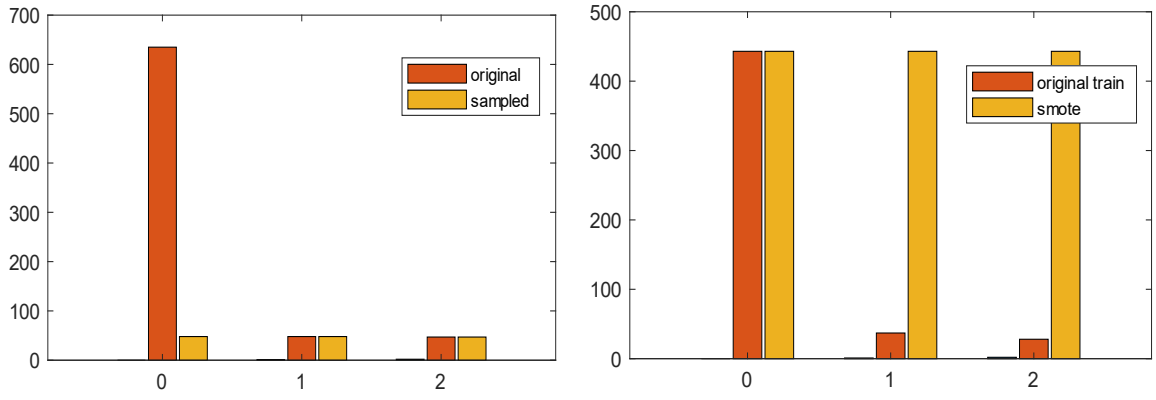
The data in Table 2 shows moderate climatic conditions with relatively low precipitation and moderate temperatures. The risk of developing Powdery Mildew is generally low, although periods of higher risk require attention.

The risk of Powdery Mildew varies by 10% on average during the day under favorable conditions [20]. Thus, it was decided to aggregate the hourly measurement results to a daily resolution. Additionally, eight attributes were introduced: 'FVT12S' – the number of hours when the temperature ranges from 12°C to 21°C; 'TL16S' – the number of hours when the temperature is below 15°C; 'TG21S' – the number of hours when the temperature exceeds 21°C; 'FVT16S' – the number of hours when the temperature ranges from 16°C to 21°C; 'TG25S' – the number of hours when the temperature exceeds 25°C; 'SR_count' – the number of hours when solar radiation was greater than 0; 'RHG85S' – the number of hours when relative humidity was greater than or equal to 85%; 'LW_count' – the number of hours when the leaves were wet.

Taking into account the above transformations, the columns of the final table are renamed according to the predefined names stored in the variable INPUT_SUMMARY_COLUMNS = ['AT_mean', 'FVT12S', 'TL16S', 'TG21S', 'FVT16S', 'TG25S', 'DP_mean', 'SR_sum', 'SR_count', 'VPD_mean', 'RH_mean', 'RHG85S', 'PR_sum', 'LW_count', 'WS_mean', 'WD_mean', 'WG_mean', 'ST_mean', 'ET_sum', 'PMBG_mean'], and the calculations correspond to the Python code (see Appendix A). The difference between the value of the disease risk for the previous day and the current day is calculated using diff(). Then, based on this difference, a new attribute 'PMBG_class' is created in which the class of risk change is stored: 1 – risk has increased, 2 – risk has decreased, 0 – risk has not changed.

The data was then split into training and test samples in a 70:30 ratio. 70% of the data was used to train the models, and 30% to evaluate their performance.

After performing these steps, the imbalance of classes in the target variable of the training dataset was detected (class 1 – 6.6%, class 2 – 6.4%, class 0 – 87%). This can lead to the model learning to favor a more common class, ignoring important features of less represented classes, which ultimately degrades the overall performance of the classifier model. Two approaches were used to address this problem: 'undersampling' and 'oversampling'. The 'undersampling' approach is about reducing the number of instances of the majority class to achieve a balance with the minority. This allows the model to better account for the minority, as the number of samples of all classes becomes proportional. To implement this approach, the pandas.DataFrame.sample() function was used, which allows randomly selecting instances from the majority class. Figure 2(a) shows a comparison of the original dataset before dividing it into training and test samples and the data after resampling. The second approach is 'oversampling', which implies increasing the number of minority samples to achieve balance with the majority. One of the most common methods for this is the Synthetic Minority Over-sampling Technique (SMOTE) [21, 22]. This method generates synthetic minority samples by interpolating between real samples. It works by selecting a few nearest neighbors for each minority pattern and creating new patterns based on these neighbors. Figure 2(b) shows a comparison of the original training set and its version after SMOTE.



(a) (b)

Figure 2: (a) comparison of original and resampled preprocessed dataset across three categories, (b) comparison of original train data and resampled train data with SMOTE across three categories.

These two approaches help to achieve a balance in the data and improve the overall quality of the model, preventing it from being biased towards the class with more instances and increasing classification accuracy for less represented classes.

The next step in the process of preparing data before training classification models is scaling, including standardization and normalization. Scaling is an important step in data processing because different features can have different scales, which can negatively affect the performance of machine learning models, especially those based on distance or gradient methods.

The mean and standard deviation are computed from the training data, and these values are then used to standardize the training data. The same mean and standard deviation from the training set are also applied to standardize the test data. This ensures that both datasets are transformed consistently, allowing for accurate evaluation of the performance of the model. The standardization is performed in MATLAB Classification Learner App [23] automatically before training the models.

The last step is to extract significant features. To do this, the 'Feature Selection Wrapper Class' algorithm from a GitHub toolbox was used [24]. In the research, the Harris Hawk Optimizer (HHO) method was used to extract significant features. This feature selection method mimics the behavior and joint hunting strategy of Harris's hawks when chasing prey.

The algorithm proposed by Heidari and co-authors [25] is based on the swarm approach, does not use gradients, and includes evolutionary optimization elements. HHO consists of two main phases, exploration and exploitation, which alternate in time to find the best parameters.

This algorithm has a high convergence rate and excellent local search capabilities, which makes it effective for feature selection problems. In this research, the HHO algorithm proposed five significant features ('TL16S', 'FVT16S', 'DP_mean', 'ST_mean', 'ET_sum'), which were extracted for further analysis and model training.

After feature selection, the training set was used to train various Simulink-compatible machine-learning classifiers. Types of classifiers chosen and their relevant hyperparameters [26, 27] are summarized in Table 3.

Table 3
Models and Hyperparameters used for training and testing

Model Type	Preset	Hyperparameters
Tree	Fine Tree	Maximum number of splits: 100; Split criterion: Gini's diversity index; Surrogate decision splits: Off
Tree	Medium Tree	Maximum number of splits: 20; Split criterion: Gini's diversity index; Surrogate decision splits: Off
Tree	Coarse Tree	Maximum number of splits: 4; Split criterion: Gini's diversity index; Surrogate decision splits: Off

Discriminant	Linear Discriminant	Covariance structure: Full
Discriminant	Quadratic Discriminant	Covariance structure: Full
Efficient Logistic Regression	Efficient Logistic Regression	Learner: Logistic regression; Solver: Auto; Regularization: Auto; Regularization strength (Lambda): Auto; Relative coefficient tolerance (Beta tolerance): 0.0001; Multiclass coding: One-vs-One
Efficient Linear SVM	Efficient Linear SVM	Learner: SVM; Solver: Auto; Regularization: Auto; Regularization strength (Lambda): Auto; Relative coefficient tolerance (Beta tolerance): 0.0001; Multiclass coding: One-vs-One
Naive Bayes	Gaussian Naive Bayes	Distribution name for numeric predictors: Gaussian; Distribution name for categorical predictors: Not Applicable
Naive Bayes	Kernel Naive Bayes	Distribution name for numeric predictors: Kernel; Distribution name for categorical predictors: Not Applicable; Kernel type: Gaussian; Support: Unbounded; Standardize data: Yes
SVM	Linear SVM	Kernel function: Linear; Kernel scale: Automatic; Box constraint level: 1; Multiclass coding: One-vs-One; Standardize data: Yes
SVM	Quadratic SVM	Kernel function: Quadratic; Kernel scale: Automatic; Box constraint level: 1; Multiclass coding: One-vs-One; Standardize data: Yes
SVM	Cubic SVM	Kernel function: Cubic; Kernel scale: Automatic; Box constraint level: 1; Multiclass coding: One-vs-One; Standardize data: Yes
SVM	Fine Gaussian SVM	Kernel function: Gaussian; Kernel scale: 0.56; Box constraint level: 1; Multiclass coding: One-vs-One; Standardize data: Yes
SVM	Medium Gaussian SVM	Kernel function: Gaussian; Kernel scale: 2.2; Box constraint level: 1; Multiclass coding: One-vs-One; Standardize data: Yes
SVM	Coarse Gaussian SVM	Kernel function: Gaussian; Kernel scale: 8.9; Box constraint level: 1; Multiclass coding: One-vs-One; Standardize data: Yes
KNN	Fine KNN	Number of neighbors: 1; Distance metric: Euclidean; Distance weight: Equal; Standardize data: Yes
KNN	Medium KNN	Number of neighbors: 10; Distance metric: Euclidean; Distance weight: Equal; Standardize data: Yes
KNN	Coarse KNN	Number of neighbors: 100; Distance metric: Euclidean; Distance weight: Equal; Standardize data: Yes
KNN	Cosine KNN	Number of neighbors: 10; Distance metric: Cosine; Distance weight: Equal; Standardize data: Yes
KNN	Cubic KNN	Number of neighbors: 10; Distance metric: Minkowski (cubic); Distance weight: Equal; Standardize data: Yes
KNN	Weighted KNN	Number of neighbors: 10; Distance metric: Euclidean; Distance weight: Squared inverse; Standardize data: Yes
Ensemble	Boosted Trees	Ensemble method: AdaBoost; Learner type: Decision tree; Maximum number of splits: 20; Number of learners: 30; Learning rate: 0.1; Number of predictors to sample: Select All

Ensemble	Bagged Trees	Ensemble method: Bag; Learner type: Decision tree; Maximum number of splits: 99; Number of learners: 30; Number of predictors to sample: Select All
Ensemble	RUSBoosted Trees	Ensemble method: RUSBoost; Learner type: Decision tree; Maximum number of splits: 20; Number of learners: 30; Learning rate: 0.1; Number of predictors to sample: Select All
Neural Network	Narrow Neural Network	Number of fully connected layers: 1; First layer size: 10; Activation: ReLU; Iteration limit: 1000; Regularization strength (Lambda): 0; Standardize data: Yes
Neural Network	Medium Neural Network	Number of fully connected layers: 1; First layer size: 25; Activation: ReLU; Iteration limit: 1000; Regularization strength (Lambda): 0; Standardize data: Yes
Neural Network	Wide Neural Network	Number of fully connected layers: 1; First layer size: 100; Activation: ReLU; Iteration limit: 1000; Regularization strength (Lambda): 0; Standardize data: Yes
Neural Network	Bilayered Neural Network	Number of fully connected layers: 2; First layer size: 10; Second layer size: 10; Activation: ReLU; Iteration limit: 1000; Regularization strength (Lambda): 0; Standardize data: Yes
Neural Network	Trilayered Neural Network	Number of fully connected layers: 3; First layer size: 10; Second layer size: 10; Third layer size: 10; Activation: ReLU; Iteration limit: 1000; Regularization strength (Lambda): 0; Standardize data: Yes

Based on the described methodology for assessing the risk of Powdery Mildew Blumeria graminis in wheat, the research steps of this article were presented in the form of a structural algorithmic scheme, as shown in Figure 3.

When training the models, the five-fold cross-validation methodology was used to assess their performance [28]. This approach allows for efficient use of available data and reduces the risk of model overtraining.

The data is divided into five subsets of equal size. At each iteration, four subsets are used to train the model, and one is used to test it. As a result, average accuracy rates are obtained, which gives a more stable and reliable assessment of model quality on different chunks of the dataset.

3. Results and discussion

After training and testing the classification models, the following results were obtained, as shown in Table 4. For each model, training was performed using undersampling and oversampling data balancing approaches, in particular, the accuracy rates on the validation (Val.) and test (Test) datasets were compared.

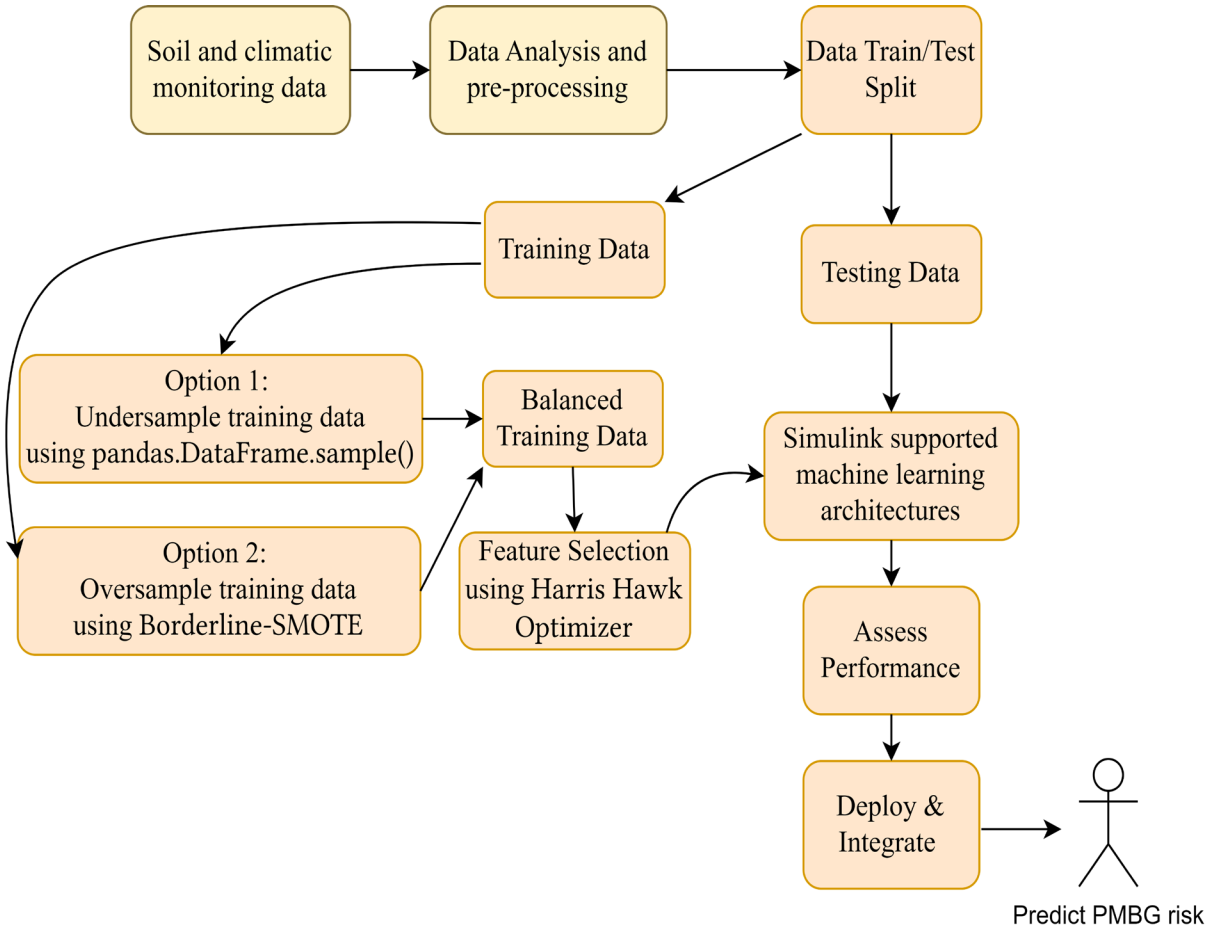


Figure 3: Machine learning pipeline to predict PMBG risk using soil and climatic monitoring data.

Table 4
The summary of the findings from the various machine learning models applied in this study with respect to the Harris Hawks algorithm

Model	Undersample, Acc. % (Val.)	Undersample, Acc. % (Test)	Oversample, Acc. % (Val.)	Oversample, Acc. % (Test)
Fine Tree	74	65	84	73
Medium Tree	74	65	83	73
Coarse Tree	74	56	76	65
Linear Discriminant	69	58	75	62
Quadratic Discriminant	60	65	82	64
Efficient Logistic Regression	69	58	74	62
Efficient Linear SVM	68	58	76	59
Gaussian Naive Bayes	65	65	72	63
Kernel Naive Bayes	76	70	76	56
Linear SVM	68	60	76	60
Quadratic SVM	72	67	87	76
Cubic SVM	70	63	90	80
Fine Gaussian SVM	69	58	91	78
Medium Gaussian SVM	73	58	84	71
Coarse Gaussian SVM	61	49	73	58
Fine KNN	72	60	90	75
Medium KNN	68	58	86	71
Coarse KNN	39	19	76	63
Cosine KNN	57	60	85	69

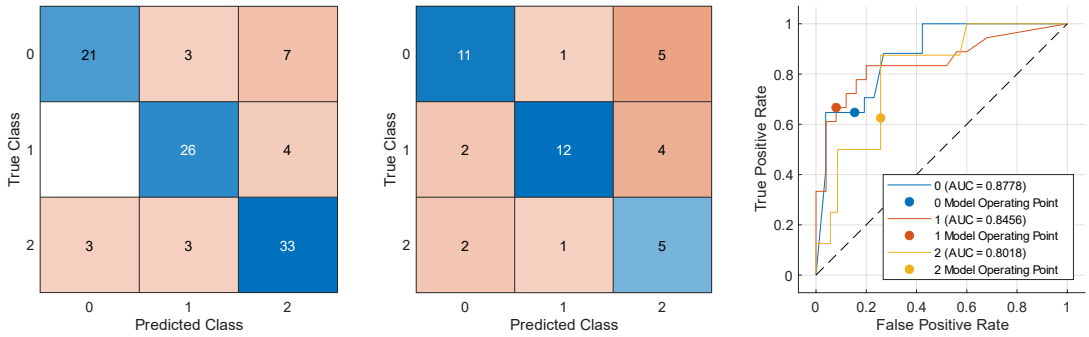
Cubic KNN	68	58	86	72
Weighted KNN	75	63	89	72
Boosted Trees	48	65	84	72
Bagged Trees	**80**	**65**	88	74
RUSBoosted Trees	71	63	83	75
Narrow Neural Network	70	63	88	73
Medium Neural Network	75	58	**91**	**80**
Wide Neural Network	69	70	90	78
Bilayered Neural Network	69	56	90	71
Trilayered Neural Network	69	60	89	76

The tree models (Fine Tree, Medium Tree, Coarse Tree) showed the best accuracy when using oversampling, with the best accuracy for Fine Tree (84% on validation and 73% on testing). Coarse Tree significantly reduced the accuracy on the undersampled test set (56%). SVM models had stable results, especially when using oversampling. For example, Quadratic SVM and Cubic SVM demonstrated high accuracy rates (up to 90% on the validation set). Fine KNN and Weighted KNN achieved the best results in oversampling, showing 90% and 89% accuracy, respectively, on validation. Medium and Wide Neural Networks showed the highest accuracy (91% and 90%, respectively) with oversampling, indicating their ability to efficiently process more balanced data. The tree-based methods (Boosted Trees, Bagged Trees, RUSBoosted Trees) also performed well, especially Bagged Trees, with 88% accuracy on the validation and 74% on the oversampled test set. Thus, the use of oversampling generally improved the performance of the models, especially for SVMs, KNNs, and neural networks.

To analyze the performance of the models in more detail, the Confusion Matrix for the training and test data, as well as the ROC curves for the three classes, were constructed, as shown in Figure 4 and Figure 5. This analysis was performed for two models: Bagged Trees (on undersampled data) and Medium Neural Network (on oversampled data).

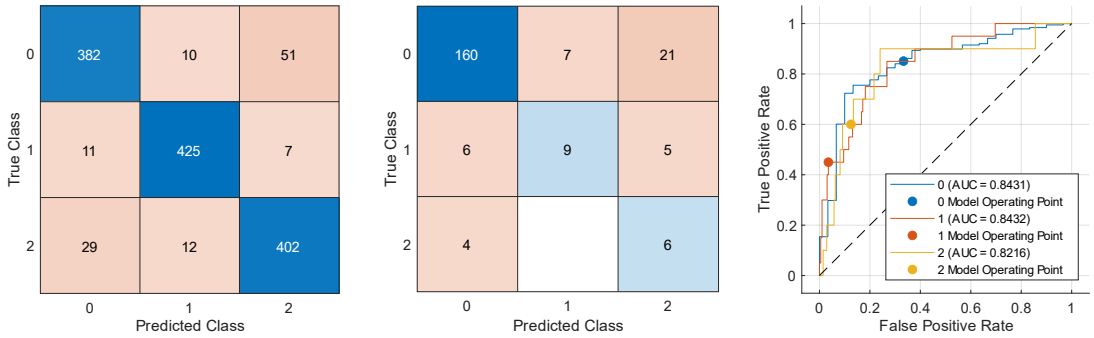
The Confusion Matrix for both models showed similar results. For the Bagged Trees model on the undersampled data, it can be seen that the model generally copes with the classification of classes, although there are some errors in the classification of smaller classes. The Medium Neural Network model, on the other hand, showed better results on oversampled data, reducing the number of misclassifications, especially for less represented classes.

The ROC curves for both models show high sensitivity and specificity for each of the three classes. Both models perform well for most classes, with the Medium Neural Network on oversampled data showing slightly higher performance in terms of area under the curve (AUC) for class '2'.



(a) (b) (c)

Figure 4: Results for Bagged Trees with undersampled data (a) Validation confusion matrix, (b) Test confusion matrix, (c) Test ROC curve.



(a) (b) (c)

Figure 5: Results for Medium Neural Network with oversampled data (a) Validation confusion matrix, (b) Test confusion matrix, (c) Test ROC curve.

The Bagged Trees model trained on undersampled data was then exported [29] to the Simulink environment (Figure 6) to implement a computerized model for automatic data processing to predict the Risk of Powdery Mildew in Wheat.

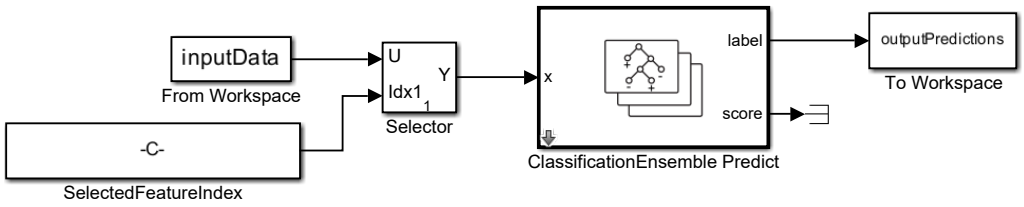


Figure 6: Simulink model for Bagged Trees with undersampled data.

An example of the model output in Figure 6 for the northern steppe scenario in the form of time graphs is shown in Figure 7, comparing actual and predicted data on the risk of Powdery Mildew Blumeria graminis in wheat.

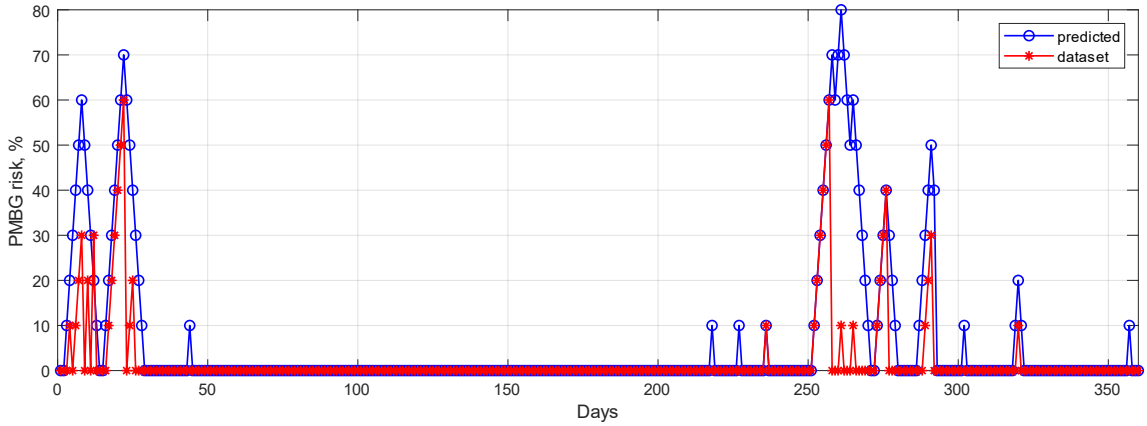


Figure 7: Results of modeling in Simulink for Bagged Trees with undersampled data for the period from September 2022 to September 2023.

The analysis of time graphs shows that the predicted data from the model is able to reproduce the main trends of growth and reduction of disease risk in the relevant time periods. Although there are some discrepancies between the actual and predicted values at certain points, in general, the model reflects risk behavior well and can be used for operational monitoring and decision-making in open-field conditions.

4. Conclusion

In this research, the relevant scientific and practical task has been solved by developing and researching the computerized method for predicting the risk of powdery mildew in wheat based on software analysis of soil and climatic monitoring data. The research allowed for the further development of information technologies for agrotechnical purposes to predict the risk of occurrence of grain crop diseases (Powdery Mildew Blumeria graminis in wheat) at the pre-symptomatic stage. The main results of the research include:

1. Using undersampling and oversampling methods to solve the problem of class imbalance in the training sample.
2. The application of feature selection algorithms, in particular Harris Hawk Optimizer, reduced the number of features to five.
3. Classification models such as Bagged Trees and Medium Neural Network performed well on both validation and test datasets, demonstrating good generalizability.
4. The export of the Bagged Trees model to the Simulink environment and the subsequent generation of program code for microcontroller devices allows it to be used for real-world prediction and control in agroclimatic systems based on fog architecture.

To improve the results, future research should pay attention to the following aspects:

1. Improving the parameters of algorithms such as Bagged Trees and Neural Networks by further tuning hyperparameters, which can lead to even more accurate results.
2. Involvement of new data sources, such as information on field cultivation, which can improve the recognition of conditions that contribute to the occurrence and development of diseases.
3. Further integration of the model with IoT platforms for automatic real-time monitoring can provide more up-to-date and accurate data for predicting disease risks.

Acknowledgments

This research was carried out as part of the scientific project 'Development of software and hardware of intelligent technologies for sustainable crop production in wartime and post-war' funded by the Ministry of Education and Science of Ukraine at the expense of the state budget (state registration number 0124U000289).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] I. Laktionov, G. Diachenko, V. Kashtan, A. Vizniuk, V. Gorev, K. Khabarlak, Y. Shedlovska, A Comprehensive Review of Recent Approaches and Hardware-Software Technologies for Digitalisation and Intellectualisation of Open-Field Crop Production: Ukrainian Case Study in the Global Context, Computers and Electronics in Agriculture 225 (2024) 1–31. doi:10.1016/j.compag.2024.109326.

[2] A. Soussi, E. Zero, R. Sacile, D. Trinchero, M. Fossa, Smart Sensors and Smart Data for Precision Agriculture: A Review, Sensors 24 (8) (2024) 1–32. doi: 10.3390/s24082647.

[3] USDA: U.S. Department of Agriculture, 2024. URL: https://www.usda.gov/.

[4] FAO: Food and Agriculture Organization of the United Nations, 2024. URL: https://www.fao.org/home/en/.

[5] FAOSTAT: Food and Agriculture Organization of the United Nations, 2024. URL: https://www.fao.org/faostat/en/#data/QCL.

[6] Q. Zheng, W. Huang, Q. Xia, Y. Dong, H. Ye, H. Jiang, S. Chen, S. Huang, Remote Sensing Monitoring of Rice Diseases and Pests from Different Data Sources: A Review, Agronomy 13 (7) (2023) 1–18. doi: 10.3390/agronomy13071851.

[7] P. Karmakar, S.W. Teng, M. Murshed, S. Pang, Y. Li, H. Lin, Crop monitoring by multimodal remote sensing: A review, Remote Sensing Applications: Society and Environment 33 (2024) 1–15. doi: 10.1016/j.rsase.2023.101093.
[8] A. San Bautista, D. Fita, B. Franch, S. Castiñeira-Ibáñez, P. Arizo, M.J. Sánchez-Torres, I. Becker-Reshef, A. Uris, C. Rubio, Crop Monitoring Strategy Based on Remote Sensing Data (Sentinel-2 and Planet), Study Case in a Rice Field after Applying Glycinebetaine, Agronomy 12 (3) (2022) 1–23. doi: 10.3390/agronomy12030708.
[9] L. Kouadio, M. El Jarroudi, Z. Belabess, S.-E. Laasli, M.Z.K. Roni, I.D.I. Amine, N. Mokhtari, F. Mokrini, J. Junk, R. Lahlali, A Review on UAV-Based Applications for Plant Disease Detection and Monitoring, Remote Sensing 15 (17) (2023) 1–23. doi: 10.3390/rs15174273.
[10] A. Abbas, Z. Zhang, H. Zheng, M.M. Alami, A.F. Alrefaei, Q. Abbas, S.A.H. Naqvi, M.J. Rao, W.F.A. Mosa, Q. Abbas, A. Hussain, M.Z. Hassan, L. Zhou, Drones in Plant Disease Assessment, Efficient Monitoring, and Detection: A Way Forward to Smart Agriculture, Agronomy 13 (6) (2023) 1–26. doi: 10.3390/agronomy13061524.
[11] S. Wang, P. Qi, W. Zhang, X. He, Development and application of an intelligent plant protection monitoring system, Agronomy 12 (5) (2022) 1–15.
[12] I. Laktionov, G. Diachenko, V. Koval, M. Yevstratiev, Computer-oriented model for Network Aggregation of Measurement Data in IoT monitoring of soil and climatic parameters of agricultural crop production enterprises, Baltic Journal of Modern Computing 11 (3) (2023) 500–522. doi: 10.22364/bjmc.2023.11.3.09.
[13] O. Kovalchuk, A Machine Learning Cluster Model for the Decision-Making Support in Criminal Justice, Computer Systems and Information Technologies 3 (2023) 51–58. doi: 10.31891/csit-2023-3-6.
[14] O. Pavlova, V. Alekseiko, The Concept of an Information System for Forecasting the Temperature Regime of the Earth's Surface Based on Machine Learning, Computer Systems and Information Technologies 2 (2024) 6–13. doi: 10.31891/csit-2024-2-1.
[15] K.G. Liakos, P. Busato, D. Moshou, S. Pearson, D. Bochtis, Machine Learning in Agriculture: A Review, Sensors 18 (8) (2018) 1–29. doi: 10.3390/s18082674.
[16] D. Tamayo-Vera, X. Wang, M. Mesbah, A Review of Machine Learning Techniques in Agroclimatic Studies, Agriculture 14 (3) (2024) 1–19. doi: 10.3390/agriculture14030481.
[17] C.E. Hachimi, S. Belaqziz, S. Khabba, B. Sebbar, D. Dhiba, A. Chehbouni, Smart Weather Data Management Based on Artificial Intelligence and Big Data Analytics for Precision Agriculture, Agriculture 13 (1) (2023) 1–22. doi: 10.3390/agriculture13010095.
[18] S.O. Araújo, R.S. Peres, J.C. Ramalho, F. Lidon, J. Barata, Machine Learning Applications in Agriculture: Current Trends, Challenges, and Future Perspectives, Agronomy 13 (12) (2023) 1–27.
[19] J. Šuljug, J. Spišić, K. Grgić, D. Žagar, A Comparative Study of Machine Learning Models for Predicting Meteorological Data in Agricultural Applications, Electronics 13 (16) (2024) 1–20.
[20] J. Bradley, G. Thomas, Wheat powdery mildew epidemiology and crop management options. In: GRDC Update papers, 2019. URL: https://grdc.com.au/resources-and-publications/grdc-update-papers/tab-content/grdc-update-papers/2019/02/wheat-powdery-mildew-epidemiology-and-crop-management-options.
[21] I. Dey, V. Pratap, A Comparative Study of SMOTE, Borderline-SMOTE, and ADASYN Oversampling Techniques using Different Classifiers. 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, (2023) 294–302.
[22] SMOTE – Synthetic Minority Over-sampling Technique. URL: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html#id1.
[23] Classification Learner App, 2024. URL: https://www.mathworks.com/help/stats/classification-learner-app.html.
[24] J. Too, Jx-WFST: A Wrapper Feature Selection Toolbox, 2024. URL: https://github.com/JingweiToo/Wrapper-Feature-Selection-Toolbox.

[25] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems, 97 (2019) 849-872. doi: 10.1016/j.future.2019.02.028.
[26] Train Classification Models in Classification Learner App. URL: https://www.mathworks.com/help/stats/train-classification-models-in-classification-learner-app.html.
[27] M. Patlak, M. Çunkaş, U. Taskiran, The Innovative Approach to Real-Time Detection of Fuel Types Based on Ultrasonic Sensor and Machine Learning. Arabian Journal for Science and Engineering 49 (2024) 16571–16591. doi: 10.1007/s13369-024-09092-5.
[28] K. Pal, B.V. Patel, Data Classification with k-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques. In: 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, (2020) 83–87. doi: 10.1109/ICCMC48092.2020.ICCMC-00016.
[29] Export Classification Model to Predict New Data, 2024. URL: https://www.mathworks.com/help/stats/export-classification-model-for-use-with-new-data.html.

Appendix A. Python code for daily summarized data

```
def read_df_with_daily_summary(csv_path: str) -> pd.DataFrame:
    df = pd.read_csv(csv_path)
    df["DT"] = pd.to_datetime(df["DT"], dayfirst=True)
    df.set_index("DT", inplace=True)
    # Resample data by day to calculate the required values
    df_daily_summary = df.resample("D").agg(
        {
            "AT": [
                "mean",
                lambda x: ((x >= 12) & (x <= 21)).sum(),
                lambda x: (x < 15).sum(),
                lambda x: (x > 21).sum(),
                lambda x: ((x >= 16) & (x <= 21)).sum(),
                lambda x: (x >= 25).sum(),
            ],
            "DP": "mean",
            "SR": ["sum", lambda x: (x > 0).sum()],
            "VPD": "mean",
            "RH": ["mean", lambda x: (x >= 85).sum()],
            "PR": "sum",
            "LW": lambda x: (x > 0).sum(),
            "WS": "mean",
            "WD": "mean",
            "WG": "mean",
            "ST": "mean",
            "ET": "sum",
            "PMBG": "mean",
        }    )
    df_daily_summary.columns = INPUT_SUMMARY_COLUMNS
    df_daily_summary["PMBG_class"] = (
        df_daily_summary["PMBG_mean"]
        .diff()
        .apply(lambda x: 1 if x > 0 else 2 if x < 0 else 0)
    )
    # Reset the index for a clean DataFrame
    df_daily_summary.reset_index(inplace=True)
    df_daily_summary.drop(columns=["DT", "PMBG_mean"], inplace=True)
    return df_daily_summary
```