applied sciences

MDPI

Article

Machine Learning Classification–Regression Schemes for Desert Locust Presence Prediction in Western Africa

L. Cornejo-Bueno [1,*], J. Pérez-Aracil [1], C. Casanova-Mateo [2], J. Sanz-Justo [3] and S. Salcedo-Sanz [1]

[1] Department of Signal Processing and Communications, Universidad de Alcalá, 28805 Alcalá de Henares, Spain; jorge.perezaracil@uah.es (J.P.-A.); sancho.salcedo@uah.es (S.S.-S.)
[2] Department of Information Systems, Universidad Politécnica de Madrid, 28031 Madrid, Spain; carlos.casanova@upm.es
[3] Laboratorio de Teledetección (LATUV), Remote Sensing Laboratory, Universidad de Valladolid, 47002 Valladolid, Spain; julia.sanz.justo@uva.es
* Correspondence: laura.cornejo@uah.es

Abstract: For decades, humans have been confronted with numerous pest species, with the desert locust being one of the most damaging and having the greatest socio-economic impact. Trying to predict the occurrence of such pests is often complicated by the small number of records and observations in databases. This paper proposes a methodology based on a combination of classification and regression techniques to address not only the problem of locust sightings prediction, but also the number of locust individuals that may be expected. For this purpose, we apply different machine learning (ML) and related techniques, such as linear regression, Support Vector Machines, decision trees, random forests and neural networks. The considered ML algorithms are evaluated in three different scenarios in Western Africa, mainly Mauritania, and for the elaboration of the forecasting process, a number of meteorological variables obtained from the ERA5 reanalysis data are used as input variables for the classification–regression machines. The results obtained show good performance in terms of classification (appearance or not of desert locust), and acceptable regression results in terms of predicting the number of locusts, a harder problem due to the small number of samples available. We observed that the RF algorithm exhibited exceptional performance in the classification task (presence/absence) and achieved noteworthy results in regression (number of sightings), being the most effective machine learning algorithm among those used. It achieved classification results, in terms of F-score, around the value of 0.9 for the proposed Scenario 1.

Keywords: desert locusts; classification; regression; machine learning methods

Citation: Cornejo-Bueno, L.; Pérez-Aracil, J.; Casanova-Mateo, C.; Sanz-Justo, J.; Salcedo-Sanz, S. Machine Learning Classification–Regression Schemes for Desert Locust Presence Prediction in Western Africa. *Appl. Sci.* **2023**, *13*, 8266. https://doi.org/10.3390/app13148266

Academic Editor: Zhengjun Qiu

Received: 16 May 2023
Revised: 21 June 2023
Accepted: 5 July 2023
Published: 17 July 2023

1. Introduction

The desert locust (*Schistocerca gregaria* sp.) is considered one of the most dangerous migratory pest species in the world [1,2], due to its fast reproduction time, capability of adaptation to and survival in difficult conditions [3], ability to migrate long distances and its harmful impact on crops [4]. Specifically, plagues of desert locusts have swept the Western Africa region for centuries [5], with high impacts in terms of crops losses and food scarcity, on the environment (due to pesticides and treatments against these insects [6]) and finally on the economy of the affected countries [7]. Countries such as Mauritania, Burkina Faso, Mali or Niger have suffered devastating desert locust plagues in recent decades, such as those between winter 2003 and spring 2004 in this zone [1].

The Food and Agriculture Organization of the United Nations (FAO) [8] has pointed out the necessity of implementing preventive strategies in the main affected countries. Western and Eastern Africa [5,9], Southwest Asia [10] and the area around the Red Sea are the zones with favourable climatic conditions to witness an upsurge of desert locust plagues [1]. These preventive strategies include the detection of breeding ground areas, the tracking of locust intrusion with Internet of Things devices [11] and the treatment

with pesticides whenever possible, in order to reduce the probability of plagues in a given region [12]. There are many previous studies on the detection of desert locust breeding ground areas, using different data (satellite [5,13,14] and ground measurements [15]) and including modern computational techniques [16].

In recent years, the use of machine learning (ML) algorithms has been very important in Earth observation problems [17–19]. In ref. [20], a random forest algorithm is applied to the problem of locating desert locust juvenile individuals in Mauritania using soil moisture data. In ref. [21], a rule generation algorithm by means of genetic algorithms is proposed for the problem of predicting desert locust presence (breeding area locations). In ref. [22], a Hidden Markov Model is applied to predict the severity of locust plagues in croplands using remote sensing data. In ref. [23], different ML classifiers such as the gradient boosting, RF and Maximum Entropy methods are compared in a problem of desert locust detection in different countries of Western Africa, such as Niger, Mauritania, Mali, Algeria and Morocco. The results obtained have shown good results when the database are completed with "pseudo-absenses" data. In ref. [24], different ML classification algorithms have been applied to the problem of desert locus breeding area prediction across Africa and Asia. Specifically, eXtreme gradient boosting, Weighted k-Nearest Neighbors, Feed-Forward Neural Networks, Support Vector Machines and RF algorithms have been applied and discussed in different breeding area locations, including pseudo-absence data. In ref. [25], the problem of desert locust presence prediction using an ML classification approach (Support Vector Machines with sliding windows) and remote sensing data is tackled. The region of with pseudo-absences is considered. In ref. [26], a deep learning approach including convolutional neural networks and LSTM networks is presented for the problem of desert locust detection (classification task) in East Africa. In ref. [27], a convolutional neural network has been applied for the detection of two species, African migratory locust (*Locusta migratoria migratoriodes*) and Red locust (*Nomadacris septemfasciata*), that are prevalent in the study area in Zambia. Images taken in situ are used to train and validate the results of locust detection in that work.

All these previous approaches dealing with ML algorithms for desert locust prediction and related problems share some common features. First, all of them are focused on different regions of Africa, and some of them include data from Asia and the Middle East, using data from the FAO desert locust database [8] as the main target source of data. Second, all these previous approaches have formulated the prediction problems as classification tasks, in many cases as binary classification problems (appearance or non-appearance of locusts or locust breeding areas, etc.). In all the problems reviewed, the small number of registers in the database and the scarce observations present important problems, which have been attempted to be solved by including "pseudo-absences" [23], an attempt to extend the available databases with synthetic data, such as in [20,23]. However, this method may introduce biases into the prediction models, as pointed out by some authors [28]. Finally, all the works previously described use similar predictive variables, related to weather variables and soil moisture data.

In this paper, an alternative methodology to tackle desert locust prediction problems is proposed. For this purpose, classification–regression tasks are considered (and not only classification approaches) with ML schemes. The problem is focused on the Western Africa region, and the presence-only data are considered in three large breeding zones, in which different ML algorithms are applied to reanalysis and NDVI data. Firstly, classification approaches are applied in order to predict the presence/non-presence of locusts in each zone, in a monthly time horizon. Secondly, a regression step is addressed, in which the number of monthly visualizations of desert locusts in each zone is predicted. The results obtained show that a successful joint prediction of the occurrence and number of locust visualization events is possible with ML classification–regression schemes. The main contributions of this work can be summarized as follows:

- An ML-based classification–regression approach is proposed to deal with the problem of desert locust detection in Western Africa.

- The classification task involves a binary detection problem (yes/no) based on weather variables from reanalysis and NDVI data.
- A regression task which deals with the number of locust sightings in each study zone is proposed, using similar predictive variables to those of the classification problem.
- Experiments based on real sighting data from Western Africa have shown the potential of the proposed classification–regression approach.

The rest of the paper has been structured in the following way: The next section describes the methodology proposed, including the data available, the predictive variables and a summary of the ML algorithms considered. Section 3 presents the experiments carried out and the results obtained, with a discussion on different aspects of the application of ML classification–regression algorithms to the problem at hand. Section 4 closes the paper with some final remarks on the research carried out.

2. Methodology

In this paper, we consider an estimation problem to determine the presence or non-presence of locusts, as well as their number, in different scenarios. For this purpose, we have used time series of sightings as the target variable of the problem. These time series cover the years 2000 to 2015, and the geographical area of the study is West Africa, with Mauritania being the most interesting due to the large number of sightings. Presence data were obtained by downloading from the Locust Hub Initiative provided by the FAO (https://locust-hub-hqfao.hub.arcgis.com/). From this data source, we were able to download the presence or absence of different types of locusts (hoppers, swarms and adults). Since in the time series of locusts there is no regular temporal or spatial continuity, it was decided to segment the problem as follows: In Figure 1, we can see the presence of locusts in green color. The geographical area under study has been divided into 3 regions, about 300 km in diameter, thus forming the three classification–regression scenarios addressed in the paper. In each of these scenarios, the forecasting process consists of monthly estimation of the presence and number of locusts. The exact number of sightings on the time series used, for each scenario, is as follows: Scenario 1 = 5414 (there are 5414 sightings in 123 days); Scenario 2 = 47,131 (there are 47,131 sightings in 210 days) and Scenario 3 = 8897 (there are 8897 sightings in 103 days). To obtain a good estimation, the forecast variables used were obtained from ERA5, the fifth-generation European Centre for Medium-Range Weather Forecasting (ECMWF) reanalysis constructed from observations and atmospheric models. These reanalysis data cover the period from 1959 to present, and their update frequency is every hour [29]. As mentioned above, the estimation is monthly and from 2000 to 2015, so the time series will be of length 192 (sixteen years times twelve months), so the reanalysis variables will be averaged to monthly data. In each study region, for Scenarios 1, 2 and 3, we downloaded the reanalysis meteorological variables at 5 geographical points, and for each point we obtained a total of 7 variables, listed below: surface pressure, total column water, total column water vapour, 10 m U wind component, 10 m V wind component, 2 m temperature and skin temperature. This makes a total of 35 exogenous reanalysis variables in each area for the regression–classification process. In addition, in each zone we have an extra variable, the Normalized Difference Vegetation Index (NDVI), because of the high correlation it has with the presence or absence of locusts. The selection of these variables for the process of predicting the presence of the Mauritanian locus is based on their relevance and potential relationship with the behavior and distribution of the locust. For example,

- Surface pressure can influence weather patterns and the formation of high- and low-pressure systems that may affect the presence and movement of the locust.
- Total column water and total column water vapor are related to the availability of moisture in the atmosphere, which can be an important factor for the development and migration of the locust.
- The 10 m U and V wind components are relevant for understanding wind patterns and their influence on the dispersion and movement of the locust.

- The 2 m temperature and skin temperature can affect the activity and behavior of the locust as its metabolism and movements may be related to thermal conditions.
- NDVI provides information about vegetation and can be associated with food availability for the locust.

Overall, these variables have been selected because they are considered to capture relevant climatic, environmental and ecological aspects related to the behavior and distribution of the Mauritanian locust. The bibliographic references that can relate our study to the use of these climatic variables are [21,26,30].

A summary of the variables used and their units can be found in Table 1. In each considered zone, we carried out data partitioning of the dataset by dedicating 70% to the training set and the remaining 30% to the test set. This partitioning is kept the same, with the same samples, for all the ML models tested, in order to proceed with a fair comparison among methods. Note that all the error metrics results are obtained in the test set.

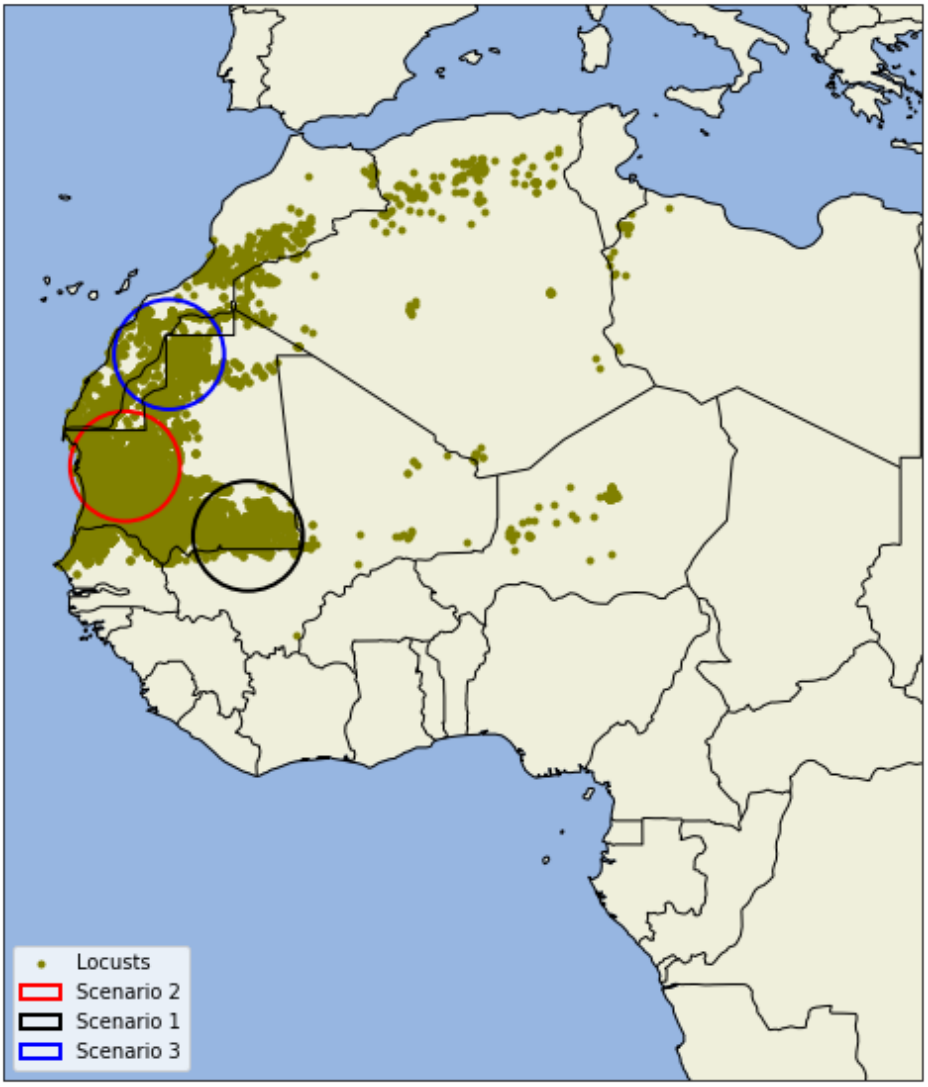


Figure 1. Geographical location map of the locust sightings, showing the different study scenarios.

Table 1. Description of the predictive (input) variables in the database for locust presence estimation, Western Africa.

Variable	Units
Surface pressure	hPa
Total column water	kg/m^2
Total column water vapour	kg/m^2
10 m U wind component	m/s
10 m V wind component	m/s
2 m temperature	K
Skin temperature	K
NDVI	-

2.1. ML Algorithms Considered

Below is a brief overview of the algorithms employed in the prediction process. It is worth noting that most of these algorithms have been utilized for both classification and regression tasks, which will be further elaborated on in Section 3.

2.1.1. Linear Regression (LR)

The linear regression method is a mathematical model utilized to determine the relationship between a target variable, $\mathbf{y}$, and a set of input variables, $\mathbf{x} = [x_1, x_2, \ldots, x_m]$ [31]. This relationship can be expressed mathematically as a linear combination of input variables, weighted by certain parameters:

$$\begin{aligned} \mathbf{y}(x, \beta) &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon \\ &= \beta_0 + \sum_{j=1}^{m} \beta_j x_j + \varepsilon \end{aligned} \tag{1}$$

where $\beta_j, \forall; j \in 0, 1, \ldots, m$ represent the parameters of the multiple regression model, and ε denotes the error associated with the prediction. Typically, the least squares method is employed to calculate the parameters of the model [32].

However, it is important to acknowledge that this approach has its limitations. Merely observing that two variables exhibit similar growth or decline patterns does not necessarily imply a causal relationship, as a spurious connection might exist between them. To address this, non-linear functions are often employed to achieve a more accurate approximation to reality. The following algorithms aim to accomplish precisely that.

2.1.2. Decision Trees (DT)

Decision trees are a popular non-parametric machine learning method utilized for both classification and regression tasks. One of their notable advantages, distinguishing them from neural networks, is their interpretability. Decision trees offer easily understandable decision rules, in contrast to the complex weight connections found in neural networks [33]. A decision tree is constructed using a combination of nodes and branches, each serving a specific purpose:

- The internal nodes or leaves represent the input predictor variables used for decision making.
- Each branch represents a decision based on a particular condition or the probability of an event occurring.
- The terminal nodes provide the final decision or outcome.

The construction of a decision tree follows a "divide and conquer" approach [34]. Given a set of k classes, $[C_1, C_2, \ldots, C_k]$, as well as a training set, $\mathbf{x} = [x_1, x_2, \ldots, x_m]$, the methodology is based on the following principles [35]:

- If the training set $\mathbf{x}$ contains elements that belong exclusively to a single class C_j, the decision tree terminates at a leaf node that indicates class C_j.

- If $\mathbf{x}$ contains features that could correspond to multiple classes, a feature is chosen for testing. This feature has associated labels or possible outcomes $[y_1, y_2, \ldots, y_n]$. The training set $\mathbf{x}$ is then divided into smaller subsets $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where each subset $\mathbf{x}_i$ contains the features from $\mathbf{x}$ associated with the chosen test label y_i. This recursive process is applied to each subset.

In the traditional methodology of constructing decision trees, the key aspect lies in how the dataset is split. The selection of a feature test determines how training objects are distributed into subsets, facilitating the construction of subtrees accordingly [34]. Notable advancements in and variations of this algorithm include C4.5, which employs information theory to evaluate decision tree splits [36].

2.1.3. Random Forest (RF)

The random forest approach, first introduced by Leo Breiman, has emerged as a significant competitor to boosting algorithms [37]. It comprises a collection of classifiers/regressors based on decision trees, often referred to as "base learners" denoted as $h_1(x), h_2(x), \ldots, h_J(x)$. Each decision tree depends on a random subset of input variables, with the same distribution applied across all trees in the forest.

In the context of a random forest, the goal is to find a predictor function, denoted as $f(\mathbf{x})$, that closely estimates the actual response given a random set of input variables $\mathbf{x} = [x_1, x_2, \ldots, x_m]$ and the corresponding response values $\mathbf{y}$. To achieve this, a loss function is employed to minimize the expected value of the loss [38]. The set of learners are then combined to yield the final $f(\mathbf{x})$, which is computed differently depending on whether it is a classification problem (Equation (2)) or a regression problem (Equation (3)):

For classification:

$$f(\mathbf{x}) = \text{mode } h_1(\mathbf{x}), h_2(\mathbf{x}), \cdots, h_J(\mathbf{x}) \tag{2}$$

For regression:

$$f(\mathbf{x}) = \frac{1}{J} \sum_{j=1}^{J} h_j(\mathbf{x}) \tag{3}$$

In both cases, the combination of the individual base learners allows for improved prediction accuracy and robustness in handling complex data patterns. Indeed, a solid understanding of decision trees is crucial for building the random forest algorithm optimally. Decision trees serve as the fundamental building blocks within the random forest framework. Each decision tree in the random forest contributes to the overall predictive power of the algorithm.

2.1.4. Support Vector Regression (SVR)

The SVR approach, a machine learning algorithm widely used in regression problems [39], has its origin in Support Vector Machines (SVM), which are considered a binary classification method with high generalization capability in high-dimensional problems [40]. While historically SVMs have primarily been utilized for binary classification problems, their applicability has expanded in recent decades to cover multiclass problems and regression tasks, leading to the development of Support Vector Regressors (SVRs) [41]. Both SVMs and SVRs revolve around constructing a regressor hyperplane that best fits the training dataset. In the case of regression, a tolerance margin is introduced between the samples and the hyperplane, ensuring that the predicted data lie within a specified margin distance. Consequently, the objective becomes selecting a hyperplane with the maximum margin between the support vectors within the dataset. Notably, certain samples may fall outside the tolerance margin and are considered errors, prompting the calculation of their

distance from the nearest boundary, denoted as ε [41]. To train the model, an optimization problem needs to be solved based on the following equation:

$$min\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad (4)$$

In the equation, ω represents the magnitude of the hyperplane, while C is a positive constant that determines the tolerance for deviations beyond the support vectors. It is important to note that C should be greater than 0. A high value of C indicates a stricter representation of the dataset by the defined hyperplane, while a small value of C allows for a larger number of errors to be permitted by the model. The effectiveness of the algorithm relies on properly selecting and implementing the C parameter as well as the chosen kernel [42].

2.1.5. Multilayer Perceptron (MLP)

The MLP algorithm is one of the most widely used machine learning tools for both regression and classification problems. It appeared in the 1980s to overcome the problem of the simple perceptron, and therefore the impossibility of learning classes of nonlinearly separable functions; unlike it, networks with an indeterminate number of hidden layers can be constructed. It is a type of artificial neural network with forward propagation, i.e., no neural output constitutes an input for the neurons of the same or previous layers. The main advantage over its predecessor is that the inclusion of a single level of hidden neurons is sufficient for the network to act as a universal function approximator [31,43]. As mentioned above, in the MLP algorithm we can find an indetermined number of hidden layers. These are placed consecutively and consist of neurons that are connected to each other through weighted links, identified in the network as ω (Figure 2). The learning process will consist of finding those values of ω that make the error between the output given by the MLP and the expected output as small as possible. The learning method normally employed is Stochastic Gradient Descent (SGD) or the backpropagation algorithm, which can be found in detail in [44]. However, other alternatives are used, such as the Levenberg–Marquardt algorithm, applied in the experiments of this study [45].

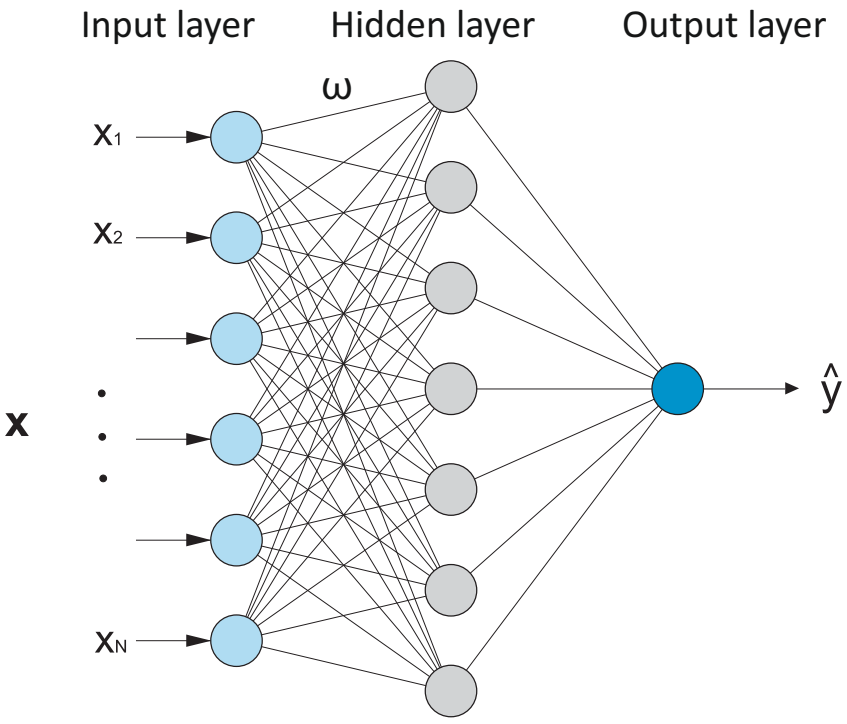


Figure 2. Multilayer perceptron structure with one hidden layer.

2.1.6. Synthetic Minority Over-Sampling Technique (SMOTE)

SMOTE is a popular data augmentation technique used in machine learning, specifically for addressing class imbalance problems [46]. In many real-world classification problems, the dataset may have imbalanced class distributions, where the minority class has significantly fewer instances than the majority class. This class imbalance can pose challenges for machine learning algorithms, as they tend to be biased towards the majority class, leading to poor predictive performance for the minority class. SMOTE addresses this issue by generating synthetic examples for the minority class, effectively oversampling the minority class to balance the class distribution. The technique works by creating synthetic samples that are interpolations between the feature vectors of neighboring instances from the minority class.

3. Experiments and Results

This section presents the experiments carried out, as well as the classification and regression results obtained for the locust sighting problem. In this section, we will specify which of the algorithms described in Section 2 have been used for regression and which for classification, and we will analyze the performance of all machines for the different scenarios. Please note that all the values shown in the results tables, for both classification and regression, have been obtained from the test set as the average of 10 runs for each of the algorithms. In addition, given the unbalanced nature of the database, the SMOTE technique has been used in the classification algorithms in order to balance the number of samples in the majority and minority classes. The difference in the results between using SMOTE and not using it in all classification cases will also be analyzed. Table 2 shows the parameters used in the ML algorithms tested in this paper.

Table 2. Hyper-parameter values for every ML model considered in the experiments.

Model	Hyper-Parameter	Meaning	Values
DT	criterion	Function to measure the quality of a split	squared error
	maxDepth	Maximum allowed depth for trees	unbounded
	min_samples_split	Number of samples required to split an internal node	2
RF	N	Number of weak learners (trees)	100
	maxDepth	Maximum allowed depth for trees	unbounded
SVR	K	Kernel	sigmoid
	γ	Kernel coefficient	$\frac{1}{(n\_samples \cdot variables)}$
	C	Regularization parameter	1
	ϵ	No penalty associated with points predicted	0.1
MLP	N	Number of hidden layers	3
	$\tilde{N}$	Number of neurons per layer	[50, 30, 10]
	activation	Activation function for the hidden layer	relu
	max_iter	Maximum number of iterations	3000
	learning_rate_init	The initial learning rate used	0.001

3.1. Classification and Regression Metrics

In this paper, we are dealing with a classification and regression problem, so it is necessary to analyze the metrics required in each case. We will begin with a description of the classification metrics used in this study, whose scalar and graphical analysis allows us to perform a correct interpretation for the evaluation of the different models [47]. We consider four scalar metrics: the area under the ROC curve (AUC), precision, recall and F-score (sometimes known as F1-score). In some cases, we will add the ROC curve as a graphical interpretation of the results. A brief description of these metrics is as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{F-score} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{7}$$

where TP is the number of true-positive samples, FP is the number of false-positive samples and FN is the number of false-negative samples. Therefore, the precision metric indicates the number of positive samples that were correctly classified with respect to the total number of predicted positive samples (true and false). The recall metric expresses the ratio of correctly classified positive samples with respect to the false-negative and true-positive samples. We must pay special attention to this metric, since a false negative (in our case, suggesting that there was no locust presence when in fact there was) may imply greater losses than false positives. Finally, the F-score metric is the harmonic mean of precision and recall. Analyzing the conclusions derived from the precision and recall metrics, a low F-score value caused by a low recall may imply a worse classifier than if this low value were caused by a low precision.

The AUC is related to the ROC curve, so we will first define this graphical classification metric. The ROC curve is a two-dimensional graph where the parameter true-positive rate (TPR), or recall, represents the y-axis, and false-positive rate ($FPR = FP/FP + TN$, where TN is the number of true-negative samples) represents the x-axis. So, this helps us to understand the benefits, which would be given by the true positives, versus the costs, given by the false positives, of a binary classifier. The AUC is used to calculate the area under the ROC curve. The values of this parameter range from 0 to 1, so there are no classifiers with good performance with AUC values below 0.5.

For the analysis of the performance of the regression machines, we will use the following scalar metrics: the Mean Squared Error (MSE), Mean Average Error (MAE), Pearson correlation coefficient (R^2) and the relative versions of MSE and MAE (MSER and MSER):

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{8}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

$$R^2 = \frac{\sum_{i=1}^{n}(y_i - \text{E}[y])(\hat{y}_i - \text{E}[\hat{y}])}{\sqrt{\sum_{i=1}^{n}(y_i - \text{E}[y])^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \text{E}[\hat{y}])^2}} \tag{10}$$

$$\text{MSER} = \frac{\frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(\bar{y})^2} \tag{11}$$

$$\text{MAER} = \frac{\frac{1}{N}\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\bar{y}}, \tag{12}$$

where $\hat{y}_i$ represents the predicted output of the regressor for the sample i, and y_i is the real value or expected value for the sample i. According to these metrics, a good performance of the regressor is demonstrated by MSE and MAE values close to zero, as well as R^2 values close to 1. Thus, the R^2 metric gives an idea of how correlated the regressor output and the real value are.

3.2. Results and Discussion

We begin by analyzing the results obtained by the classification machines. The ones considered in this study are SVM, DT and RF, some of the most successful existing classification approaches. Table 3 shows the results obtained for each of these machines, for the three scenarios considered, without applying the SMOTE balancing technique.

Table 3. Performance of the evaluated ML classification methods in the estimation of locust presence without applying SMOTE (Scenarios 1, 2 and 3).

	Methods	AUC	Precision	Recall	F-Score
Scenario 1	SVM	0.87	0.87	0.87	0.87
	DT	0.87	0.87	0.87	0.87
	RF	**0.95**	**0.91**	**0.95**	**0.93**
Scenario 2	SVM	0.73	0.74	0.73	0.72
	DT	0.66	0.67	0.66	0.66
	RF	**0.83**	**0.84**	**0.83**	**0.82**
Scenario 3	SVM	0.57	0.70	0.57	0.63
	DT	0.60	0.56	0.60	0.58
	RF	**0.75**	**0.76**	**0.75**	**0.75**

As can be seen, the best results have been obtained by the RF algorithm in all scenarios, among all ML techniques used. We should highlight those obtained for Scenario 1, with an F-score value of 0.93. In this case, a value of 0.95 has been obtained for the recall metric, and a value of 0.91 for precision. This indicates that we are dealing with a good classifier, with a low false-negative rate, which in this scenario has a higher false-positive rate which causes a lower precision. This seems to be the reason for the low F-score, and not the recall. This classifier therefore allows us to better detect the presence of locusts, with fewer errors. Proof of this is also that the best AUC value obtained, in this case 0.95, is very close to 1, which gives us an idea about the high rate of true positives and the low rate of false positives provided by this classifier. We can also highlight the importance of the use of the meteorological variables chosen for the forecasting process when using RF, noting the influence they can have on the appearance of the Mauritanian locust. As for Scenarios 2 and 3, we see that the worst results are obtained for Scenario 3, with an F-score of 0.75 and a precision and recall of 0.76 and 0.75, respectively. This may be due to the fact that this is the scenario with the lowest number of days of sightings, with a total of 103, compared to Scenario 1 and 2 with 123 and 210, respectively. This implies that the data distribution and how it correlates with the predictor variables are of utmost importance since, even though it has the lowest number of days of sightings, it does not have the lowest number of samples. In all scenarios, SVM and DT obtain the worst results, which may indicate that the meteorological variables used, as well as the number of samples, are not sufficient for these methods to obtain good results in the classification process.

Table 4 shows the results obtained by all the classification algorithms used when the SMOTE technique is applied in the pre-processing of the data. Again, the best results are obtained by the RF algorithm for all scenarios. It can be seen that for both Scenarios 1 and 2, the use of the SMOTE technique does not result in a significant improvement, but rather the opposite, going from an F-score value of 0.93 to 0.90 in Scenario 1, and from 0.82 to 0.81 in Scenario 2, in both cases using the RF algorithm. This is not a significant worsening, but this pre-processing method does not contribute to improving the performance of the RF machine. However, the situation in Scenario 3 is different: the SMOTE technique contributes to improving the results of AUC, precision, recall and F-score with values that go from 0.75, 0.76, 0.75 and 0.75, respectively, without the use of SMOTE, to 0.85, 0.77, 0.85 and 0.81 for AUC, precision, recall and F-score, respectively, when the SMOTE technique is applied. This may be due to the fact that, being the scenario with the smallest number of days of sightings, but not the scenario with the smallest number of samples, the use of a technique that involves the creation of synthetic samples favors the best performance in the learning machines. It may be because the dataset allows for better generalization and for the underlying data structure to be captured. This facilitates SMOTE in generating synthetic samples that closely resemble real instances of the minority class, thereby enhancing the model's ability to accurately recognize and classify minority instances. For the rest of the classification algorithms, SVM and DT, an improvement in performance is

observed for Scenarios 2 and 3. The SVM goes from an F-score value of 0.72 in Scenario 2, without applying SMOTE, to a value of 0.73 when the SMOTE technique is used. Likewise, in Scenario 3, for the SVM, we go from an F-score value of 0.56 without SMOTE to 0.57 with SMOTE. The same occurs with the DT algorithm, with an F-score value of 0.66 in Scenario 2 without SMOTE and 0.67 with SMOTE. The same situation is found for Scenario 3, going from an F-score value without SMOTE of 0.56 to a value of 0.60 with SMOTE. The cause of this improvement in the ML algorithms seems to be due to these learning machines requiring a large number of samples, as well as other more descriptive predictive variables. By applying the SMOTE technique, we overcome in part the issue of the small number of samples, adding other synthetic ones that contribute to better learning in the algorithms. Other descriptive variables could be used to improve the results in Scenario 1, both for the SVM and DT algorithms, because in this case we have a sufficient number of samples applying the SMOTE technique, but it is not enough to obtain a better performance.

Table 4. Performance of the evaluated ML classification methods in the estimation of locust presence when applying SMOTE (Scenarios 1, 2 and 3).

	Methods	AUC	Precision	Recall	F-Score
Scenario 1	SVM	0.88	0.83	0.87	0.85
	DT	0.86	0.83	0.86	0.84
	RF	**0.92**	**0.89**	**0.92**	**0.90**
Scenario 2	SVM	0.73	0.73	0.73	0.73
	DT	0.67	0.68	0.67	0.67
	RF	**0.81**	**0.82**	**0.81**	**0.81**
Scenario 3	SVM	0.66	0.61	0.67	0.64
	DT	0.69	0.63	0.69	0.66
	RF	**0.85**	**0.77**	**0.85**	**0.81**

Once we are able to determine the presence or absence of locusts, it would be of great interest to predict the number of individuals at each sighting location. For this reason, we combine the classification part of this paper with the prediction of the number of locusts, using different ML regression algorithms. The ones used in this study are LR, SVR, DT, RF and MLP. Table 5 shows the results obtained for each of these machines for the different scenarios considered. It is possible to see that the RF algorithm is again the approach which obtains the best result in this regression problem, improving the results of the other ML approaches in all metrics considered. Important improvements in terms of accuracy are obtained by applying the RF approach versus LR, which is the worst algorithm tested in all the scenarios considered. In Scenario 1, the second-best approach is SVR, close to the result obtained by the RF algorithm. The third and fourth approaches are MLP and DT, respectively. In Scenario 2, the second-best approach is MLP, then DF and the SVR. In Scenario 3, DT obtains better result in terms of MAE and MAER, though RF is better in terms of MSE, MSER and R^2. The third-best approach is SVR in this case, and the fourth is MLP.

Table 5. Performance of the evaluated ML regression methods in the estimation of the number of locusts (Scenarios 1, 2 and 3).

	Methods	MAE	MSE	R^2	MAER	MSER
Scenario 1	LR	55.32	6352.48	−0.45	1.28	3.38
	DT	53.56	12,792.09	−1.92	1.24	6.80
	RF	**36.11**	**2335.73**	**0.47**	**0.84**	**1.24**
	SVR	38.82	5141.12	−0.17	0.90	2.73
	MLP	51.73	7995.99	−0.83	1.19	4.25

Table 5. *Cont.*

	Methods	MAE	MSE	R^2	MAER	MSER
Scenario 2	LR	157.70	56,348.88	0.45	0.73	1.20
	DT	185.55	78,636.24	0.23	0.86	1.67
	RF	**157.59**	**54,067.41**	**0.47**	**0.73**	**1.15**
	SVR	189.52	117,055.81	−0.15	0.87	2.49
	MLP	170.91	69,203.28	0.32	0.79	1.47
Scenario 3	LR	143.24	41,401.31	−0.09	1.78	6.38
	DT	**89.89**	29,965.90	0.21	**1.12**	4.62
	RF	97.58	**22,440.73**	**0.41**	1.21	**3.46**
	SVR	94.54	44,932.54	−0.18	1.17	6.93
	MLP	101.34	32,342.32	0.15	1.26	4.99

Figure 3 shows the graphical performance of the best ML approach (RF) in all the scenarios considered. It is significant that the algorithm is able to accurately locate the greatest peaks in locust presence. There are some issues due to false positives, mainly in Scenario 3, as can be seen in the figure. In Scenario 1, there is a main peak in presence, which the RF algorithm is able to locate. In Scenario 2, there are many more peaks over time, which are also located by the algorithm in an efficient way. Scenario 3 is formed by fewer samples, with important peaks, which are located, though false positives appear, as mentioned before. The scatter plot graphs reveal that the RF algorithm works well in Scenario 1. In Scenarios 2 and 3, its performance is poorer, and false-positive results can be seen specifically in Scenario 3, probably due to the small number of samples in this scenario. These results corroborate the idea that the regression problem is much more difficult than the classification task, where the lack of data has a major effect on the performance of the regressors considered. In spite of this, we have shown that it is possible to obtain a reasonable prediction of the number of locust individuals at each sighting location with ML techniques.

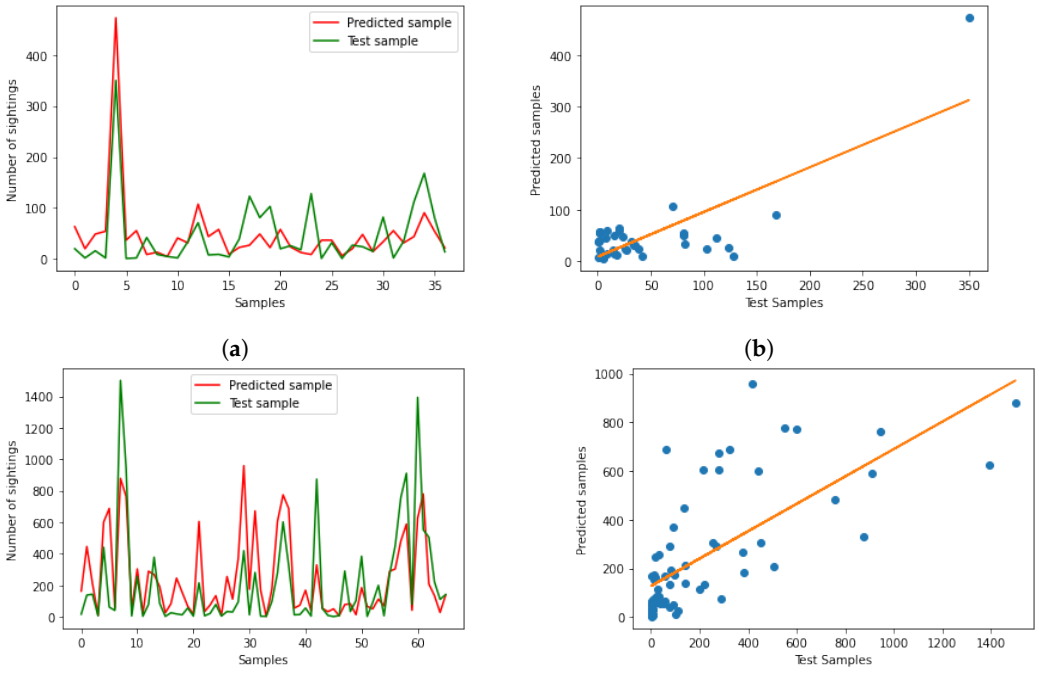
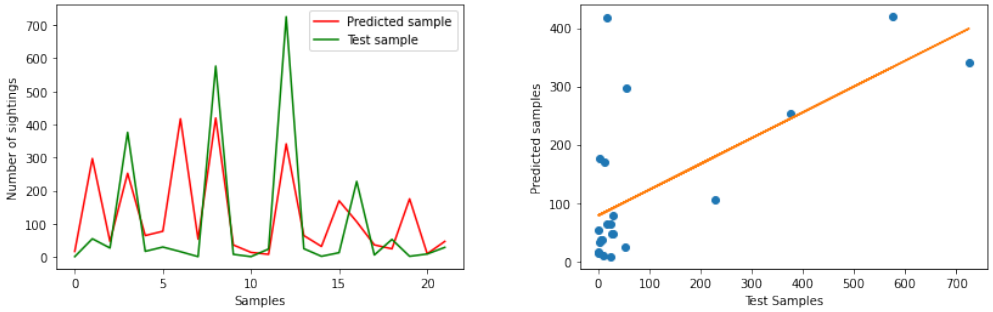


Figure 3. *Cont.*



(**e**) (**f**)

Figure 3. Performance of the best regression technique (RF) considered in the prediction problem, for Scenarios 1, 2 and 3: (**a**) temporal RF in Scenario 1; (**b**) scatter RF in Scenario 1; (**c**) temporal RF in Scenario 2; (**d**) scatter RF in Scenario 2; (**e**) temporal RF in Scenario 3; (**f**) scatter RF in Scenario 3.

4. Conclusions

In this paper, we have presented a novel way of dealing with the problem of desert locust prediction by means of a machine learning classification–regression approach. The idea is to not only deal with the presence/absence problem, but to also try to estimate the number of monthly sightings in three different scenarios for Western Africa. Three well-known classification algorithms (SVM, DT and RF) and five regression approaches have been considered (SVR, DT, RF, MLP and LR). An important number of predictive variables from reanalysis data and NDVI data are considered. We have found that the RF algorithm obtained the best results, both in classification (presence/absence) and regression (number of sightings), with excellent results in the classification task, and good results in regression, a harder problem to estimate. Some differences are spotted depending on the scenario considered, i.e., different regions of Western Africa, with different numbers of sightings in the training and test sets. In Scenario 1, the classification–regression ML approaches obtained the best results, and the worst prediction was obtained in Scenario 3 due to the smaller number of samples in this scenario.

The results obtained in this work have shown that it is possible to extend the problem of desert locust prediction to evaluating the number of monthly sightings, and not only the presence/absence as considered in previous works. In general, the performance of the ML algorithms is good in this regression problem, though in general the performance in the classification problem is better, since this is an easier approach. Future research must deal with improvements in the performance of the regression algorithms, the application of new balancing methods for the classification tasks and the analysis of the performance when a small number of samples is available, which would allow the study to be extended to other areas with fewer numbers of sightings. Also, the testing of deep learning approaches such as LSTM networks or convolutional-based approaches over similar data is a real possibility which may improve the prediction of desert locust breeding area detection in the next few years.

Author Contributions: Conceptualization, J.P.-A., S.S.-S. and L.C.-B.; methodology, J.P.-A., C.C.-M. and S.S.-S.; software, L.C.-B. and J.P.-A.; validation, J.P.-A. and S.S.-S.; investigation, J.P.-A., S.S.-S., C.C.-M. and J.S.-J.; resources, S.S.-S. and J.P.-A.; data curation, L.C.-B., C.C.-M. and J.S.-J.; writing—original draft preparation, S.S.-S., J.P.-A. and L.C.-B.; writing—review and editing, S.S.-S., J.P.-A. and L.C.-B.; supervision, J.P.-A. and S.S.-S.; project administration, S.S.-S. and J.P.-A.; funding acquisition, S.S.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the project PID2020-115454GB-C21595 of the Spanish Ministry of Science and Innovation (MICINN).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine learning
FAO	Food and Agriculture Organization of the United Nations
NDVI	Normalized Difference Vegetation Index
LR	Linear regression
DT	Decision trees
RF	Random forest
SVM	Support Vector Machine
SVR	Support Vector Regression
MLP	Multilayer Perceptron
SGD	Stochastic Gradient Descent
AUC	Area under the ROC curve
TPR	True Positive Rate
MSE	Mean Squared Error
MAE	Mean Average Error
R^2	Pearson correlation coefficient

References

1. Cressman, K. Desert locust. In *Biological and Environmental Hazards, Risks, and Disasters*; Elsevier: Amsterdam, The Netherlands, 2016; pp. 87–105.
2. Shuang, L.; Feng, S.Q.; Ullah, H.; Tu, X.B.; Zhang, Z.H. IPM-Biological and integrated management of desert locust. *J. Integr. Agric.* **2022**, *21*, 3467–3487.
3. Maeno, K.O.; Ely, S.O.; Jaavar, M.E.H.; Nakamura, S.; Ebbe, M.A.O.B. Behavioral plasticity in anti-predator defense in the desert locust. *J. Arid Environ.* **2018**, *158*, 47–50. [CrossRef]
4. Skaf, R.; Popov, G.; Roffey, J. The Desert Locust: An international challenge. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **1990**, *328*, 525–538.
5. Gómez, D.; Salvador, P.; Sanz, J.; Casanova, C.; Taratiel, D.; Casanova, J. Desert locust detection using Earth observation satellite data in Mauritania. *J. Arid Environ.* **2019**, *164*, 29–37. [CrossRef]
6. Retkute, R.; Hinton, R.G.; Cressman, K.; Gilligan, C.A. Regional Differences in Control Operations during the 2019–2021 Desert Locust Upsurge. *Agronomy* **2021**, *11*, 2529. [CrossRef]
7. Brader, L.; Djibo, H.; Faye, F.; Ghaout, S.; Lazar, M.; Luzietoso, P.; Babah, M.O. *Towards a More Effective Response to Desert Locusts and Their Impacts on Food Security, Livelihoods and Poverty*; Multilateral Evaluation of the 2003–05 Desert Locust Campaign; Food and Agriculture Organisation: Rome, Italy, 2006.
8. FAO. UN Desert Locust Program. 2022. Available online: https://www.fao.org/locusts/en/ (accessed on 19 October 2018).
9. Alemu, W.G.; Neigh, C.S. Desert Locust Cropland Damage Differentiated from Drought, with Multi-Source Remote Sensing in Ethiopia. *Remote Sens.* **2022**, *14*, 1723. [CrossRef]
10. Showler, A.T.; Shah, S.; Khan, S.; Ullah, S.; Degola, F. Desert Locust Episode in Pakistan, 2018–2021. and the Current Status of Integrated Desert Locust Management. *J. Integr. Pest Manag.* **2022**, *13*, 1. [CrossRef]
11. Dharshini, A.; Monisha, A.; Bindhu Malini, M.; Vinoth Kumar, S. Method to prevent and track Locust's Intrusion using Object Detection Algorithms. In Proceedings of the 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 10–11 March 2022; pp. 1–6.
12. Magor, J.; Lecoq, M.; Hunter, D. Preventive control and Desert Locust plagues. *Crop Prot.* **2008**, *27*, 1527–1533. [CrossRef]
13. Tratalos, J.A.; Cheke, R.A. Can NDVI GAC imagery be used to monitor desert locust breeding areas? *J. Arid Environ.* **2006**, *64*, 342–356. [CrossRef]
14. Ellenburg, W.L.; Mishra, V.; Roberts, J.B.; Limaye, A.S.; Case, J.L.; Blankenship, C.B.; Cressman, K. Detecting desert locust breeding grounds: A satellite-assisted modeling approach. *Remote Sens.* **2021**, *13*, 1276. [CrossRef]
15. Gómez, D.; Salvador, P.; Sanz, J.; Casanova, J.L. Modelling desert locust presences using 32-year soil moisture data on a large-scale. *Ecol. Indic.* **2020**, *117*, 106655. [CrossRef]
16. Villarreal, M. Desert Locusts: Can Mathematical Models Help to Control Them? In *Imagine Math 8*; Springer: Cham, Switzerland, 2022; pp. 405–417.
17. Salcedo-Sanz, S.; Ghamisi, P.; Piles, M.; Werner, M.; Cuadra, L.; Moreno-Martínez, A.; Izquierdo-Verdiguier, E.; Muñoz-Marí, J.; Mosavi, A.; Camps-Valls, G. Machine learning information fusion in Earth observation: A comprehensive review of methods, applications and data sources. *Inf. Fusion* **2020**, *63*, 256–272. [CrossRef]

18. Min, B.; Kim, T.; Shin, D.; Shin, D. Data Augmentation Method for Plant Leaf Disease Recognition. *Appl. Sci.* **2023**, *13*, 1465. [CrossRef]
19. Pentoś, K.; Mbah, J.T.; Pieczarka, K.; Niedbała, G.; Wojciechowski, T. Evaluation of Multiple Linear Regression and Machine Learning Approaches to Predict Soil Compaction and Shear Stress Based on Electrical Parameters. *Appl. Sci.* **2022**, *12*, 8791. [CrossRef]
20. Gómez, D.; Salvador, P.; Sanz, J.; Casanova, C.; Taratiel, D.; Casanova, J.L. Machine learning approach to locate desert locust breeding areas based on ESA CCI soil moisture. *J. Appl. Remote Sens.* **2018**, *12*, 036011. [CrossRef]
21. Kimathi, E.; Tonnang, H.E.; Subramanian, S.; Cressman, K.; Abdel-Rahman, E.M.; Tesfayohannes, M.; Niassy, S.; Torto, B.; Dubois, T.; Tanga, C.M.; et al. Prediction of breeding regions for the desert locust Schistocerca gregaria in East Africa. *Sci. Rep.* **2020**, *10*, 11937. [CrossRef]
22. Shao, Z.; Feng, X.; Bai, L.; Jiao, H.; Zhang, Y.; Li, D.; Fan, H.; Huang, X.; Ding, Y.; Altan, O.; et al. Monitoring and Predicting Desert Locust Plague Severity in Asia–Africa Using Multisource Remote Sensing Time-Series Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 8638–8652. [CrossRef]
23. Yusuf, I.S.; Tessera, K.A.; Tumiel, T.; Nevo, S.; Pretorius, A. On pseudo-absence generation and machine learning for locust breeding ground prediction in Africa. *arXiv* **2021**, arXiv:2111.03904.
24. Gómez, D.; Salvador, P.; Sanz, J.; Rodrigo, J.F.; Gil, J.; Casanova, J.L. Prediction of desert locust breeding areas using machine learning methods and SMOS (MIR_SMNRT2) Near Real Time product. *J. Arid Environ.* **2021**, *194*, 104599. [CrossRef]
25. Sun, R.; Huang, W.; Dong, Y.; Zhao, L.; Zhang, B.; Ma, H.; Geng, Y.; Ruan, C.; Xing, N.; Chen, X.; et al. Dynamic Forecast of Desert Locust Presence Using Machine Learning with a Multivariate Time Lag Sliding Window Technique. *Remote Sens.* **2022**, *14*, 747. [CrossRef]
26. Tabar, M.; Gluck, J.; Goyal, A.; Jiang, F.; Morr, D.; Kehs, A.; Lee, D.; Hughes, D.P.; Yadav, A. A PLAN for Tackling the Locust Crisis in East Africa: Harnessing Spatiotemporal Deep Models for Locust Movement Forecasting. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual Event, 14–18 August 2021; pp. 3595–3604.
27. Halubanza, B.; Phiri, J.; Nyirenda, M.; Nkunika, P.O.; Kunda, D. Detection of and (Orthoptera: Acrididae) MobileNet V2 Quantized Convolution Neural Network, Kazungula, Zambia. In *Cybernetics Perspectives in Systems, Proceedings of the Computer Science On-Line Conference*; Springer: Cham, Switzerland, 2022; pp. 490–501.
28. Wisz, M.S.; Guisan, A. Do pseudo-absence selection strategies influence species distribution models and their predictions? An information-theoretic approach based on simulated data. *BMC Ecol.* **2009**, *9*, 8. [CrossRef] [PubMed]
29. Hersbach, H.; Bell, B.; Berrisford, P.; Hirahara, S.; Horányi, A.; Muñoz-Sabater, J.; Nicolas, J.; Peubey, C.; Radu, R.; Schepers, D.; et al. The ERA5 global reanalysis. *Q. J. R. Meteorol. Soc.* **2020**, *146*, 1999–2049. [CrossRef]
30. Saha, A.; Rahman, S.; Alam, S. Modeling current and future potential distributions of desert locust *Schistocerca gregaria* (Forskål) under climate change scenarios using MaxEnt. *J. Asia-Pac. Biodivers.* **2021**, *14*, 399–409. [CrossRef]
31. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006; Volume 4.
32. Strejc, V. Least squares parameter estimation. *Automatica* **1980**, *16*, 535–550. [CrossRef]
33. Kotsiantis, S.B. Decision trees: A recent overview. *Artif. Intell. Rev.* **2013**, *39*, 261–283. [CrossRef]
34. Podgorelec, V.; Kokol, P.; Stiglic, B.; Rozman, I. Decision trees: An overview and their use in medicine. *J. Med Syst.* **2002**, *26*, 445–463. [CrossRef] [PubMed]
35. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
36. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
37. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
38. Cutler, A.; Cutler, D.R.; Stevens, J.R. Random forests. In *Ensemble Machine Learning*; Springer: New York, NY, USA, 2012; pp. 157–175.
39. Salcedo-Sanz, S.; Rojo-Álvarez, J.L.; Martínez-Ramón, M.; Camps-Valls, G. Support vector machines in engineering: An overview. *Data Min. Knowl. Discov.* **2014**, *4*, 234–267. [CrossRef]
40. Chapelle, O.; Haffner, P.; Vapnik, V.N. Support vector machines for histogram-based image classification. *IEEE Trans. Neural Netw.* **1999**, *10*, 1055–1064. [CrossRef]
41. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
42. Zhang, F.; O'Donnell, L.J. Support vector regression. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 123–140.
43. Haykin, S.; Network, N. A comprehensive foundation. *Neural Netw.* **2004**, *2*, 41.
44. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
45. Hagan, M.T.; Menhaj, M.B. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. [CrossRef]

46. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
47. Tharwat, A. Classification assessment methods. *Appl. Comput. Inform.* **2020**, *17*, 168–192. [CrossRef]