eng

MDPI

Article

WB Score: A Novel Methodology for Visual Classifier Selection in Increasingly Noisy Datasets

Wagner S. Billa [1,2], Rogério G. Negri [3] and Leonardo B. L. Santos [1,2,*]

[1] Center for Monitoring and Early Warning of Natural Disasters (CEMADEN), São José dos Campos 12630-000, Brazil
[2] National Institute for Space Research (INPE), São José dos Campos 12227-010, Brazil
[3] Science and Technology Institute (ICT), São Paulo State University (UNESP), São José dos Campos 12224-300, Brazil
* Correspondence: santoslbl@gmail.com

Abstract: This article addresses the challenges of selecting robust classifiers with increasing noise levels in real-world scenarios. We propose the WB Score methodology, which enables the identification of reliable classifiers for deployment in noisy environments. The methodology addresses four significant challenges that are commonly encountered: (i) Ensuring classifiers possess robustness to noise; (ii) Overcoming the difficulty of obtaining representative data that captures real-world noise; (iii) Addressing the complexity of detecting noise, making it challenging to differentiate it from natural variations in the data; and (iv) Meeting the requirement for classifiers capable of efficiently handling noise, allowing prompt responses for decision-making. WB Score provides a comprehensive approach for classifier assessment and selection to address these challenges. We analyze five classic datasets and one customized flooding dataset in São Paulo. The results demonstrate the practical effect of using the WB Score methodology is the enhanced ability to select robust classifiers for datasets in noisy real-world scenarios. Compared with similar techniques, the improvement centers around providing a visual and intuitive output, enhancing the understanding of classifier resilience against noise, and streamlining the decision-making process.

Keywords: computational classification; machine learning; noise robustness; classifier selection; visual decision-making

Citation: Billa, W.S.; Negri, R.G.; Santos, L.B.L. WB Score: A Novel Methodology for Visual Classifier Selection in Increasingly Noisy Datasets. *Eng* **2023**, *4*, 2497–2513. https://doi.org/10.3390/eng4040142

Academic Editor: Goncalo Jesus

Received: 25 July 2023
Revised: 19 September 2023
Accepted: 20 September 2023
Published: 25 September 2023

1. Introduction

Computational classification methods utilizing machine learning algorithms have gained significant popularity due to their ability to learn patterns and make accurate predictions automatically [1,2]. However, real-world scenarios often present challenges in the form of uncertainties and noise inherent in the datasets used for training and testing these classifiers [3]. This article focuses on a scenario where classifiers are trained once and deployed in production environments, continuously facing datasets with increasing noise levels over time.

When dealing with datasets prone to increasing noise, several challenges arise in selecting the most appropriate classifiers. The following problems are commonly encountered: (i) Classifiers must exhibit robustness to noise, ensuring accurate predictions despite noisy instances or attributes in the data. The selection of classifiers that can effectively handle varying levels of noise is crucial to maintaining high prediction performance over time [4]; (ii) It is often difficult to collect and label large amounts of data representative of the noise encountered in the real world. As a result, classifiers may be trained on datasets that are not representative of the noise that they will encounter in production, which can lead to poor performance [5]; (iii) Noise can take many different forms making it difficult to distinguish between noise and real data variation [6]; (iv) There are scenarios where efficient handling of real-time data streams with increasing noise requires classifiers that can process incoming

instances promptly. In cases like this, solutions like ensemble methods or active learning techniques may not be effective since retraining time can be highly costly [7].

This paper presents the novel "WB Score" methodology for selecting a robust classifier in increasingly noisy environments. The methodology inputs a dataset and a list of classifiers to test, determining the most reliable despite significant noise. By providing an intuitive graph that visually represents classifier performance, decision-makers can efficiently assess and choose an appropriate classifier based on their specific needs and the overall noise levels. Although there are graphic or visual methodologies such as robustness curves [8], noise tolerance plots [9], box plots [10], stacked bar charts [11], and heatmaps [12], none of them were designed to take into account the increase in noise nor to analyze a set of classifiers at the same time to select the best one for the task to be performed.

This paper is organized as follows: Section 2 introduces relevant related works; Section 3 presents the Data and Methods; Section 4 shows the results and relevant discussions of this analysis; finally, Section 5 contains the conclusions.

2. Related Works

As an application of machine learning, classification techniques are used to distinguish instances of a dataset into classes or groups of similar elements [13]. These techniques are widely applied in several areas of knowledge area, for example, biology [14], biometric authentication [15], computer vision [16], document classification [17], development of new drugs [18], pattern recognition [19], and natural language processing [20]. For each of these areas, there are peculiarities in their explanatory variables (or features) that make a classification algorithm better than another for a given purpose or dataset.

However, for the composition of a dataset, there may have needed to be more accurate readings in the collected values, causing errors or imprecision. This work presents a ranking method for classification algorithms that consider adding noise in the original dataset, thus enabling the choice of the most robust algorithm under this condition. The selected algorithm can then be of great value in datasets with known noisy or inaccurate information.

The "No Free Lunch theorem" [21] says that no single algorithm always performs better than others in all datasets. Depending on its data, a classifier can give the best response among all others due to some of its internal implementations, which deal better with the existing dataset bias [22]. Finding the better solution for a practical problem is formally known as the Algorithm Selection Problem in literature [23].

At first, the most logical approach for selecting a classification algorithm for a new problem is to compare the performance of previous similar ones, choosing the best option historically found. In this case, subjects like ranking [24–27], meta-learning [28,29], problem difficulty [30], and knowledge discovery [31] can also be addressed.

Another approach found is using multiple classifier systems as an alternative to improve accuracy. The idea is to have a classifier pool that can be used to compose the result, selecting the best classifier for each sample to be classified. Several studies have demonstrated its advantages over individual classification models [32].

The literature needs to include the approach of selecting the best classifier for the case in which we have a noisy dataset or is error-prone in its data acquisition [33,34]. However, some articles demonstrate that using MCS helps to reduce the effects of noise in the analyzed datasets [35–37]. The proposed methodology is a solution to fill this gap, allowing the analysis of several classifiers simultaneously to choose one that is more robust to errors or noise within the used dataset. The classifiers' accuracy response on the original dataset and the datasets in which noise is introduced are considered to find the best suitable algorithm with the used dataset.

Some works use sensitivity analysis [38,39] to identify the relationship of all fields that make up the input data, trying to find out the most important ones and discard those who contribute with a tiny fraction, which can then be considered negligible [40]. Methods like

One-Factor-at-a-Time [41] and Elementary Effects [42] are examples of this, and they are helpful in large modeled systems with dozens of inputs or more, but this work is different. Here, the objective is to check how the output response degrades from the original ones as the noise levels grow up, regardless of how the input data interact with themselves.

3. Data and Methods

The data are presented in Section 3.1, whereas the methods are explained from Section 3.2 through to Section 3.7.

3.1. Used Datasets

Five well-known classic public datasets were used in this experiment: IRIS, GLASS, IONOSPHERE, IMAGE SEGMENTATION, and SEEDS. All of them are available to download from the UC Irvine Machine Learning Repository (https://archive.ics.uci.edu (accessed on 1 July 2023)). A customized flooding dataset in São Paulo was also analyzed (FLOODINGS SP) [43]. Table 1 gives a short description of each dataset and the size of elements.

Table 1. Description of datasets used in the experiment.

Dataset	Description	Instances	Attributes
IRIS	Classification of iris flowers into three species: setosa, versicolor, and virginica.	150	4
GLASS	Classification of glass types into six categories: window, bottle, table, vehicle, laboratory, and other.	214	9
IONOSPHERE	Classification of ionospheric conditions into three categories: quiet, disturbed, and very disturbed.	351	34
IMAGE SEGMENTATION	The instances were drawn randomly from a database of 7 outdoor images and were hand segmented to create a classification for every pixel.	210	19
SEEDS	Classification of seeds into three species: setaria italica, digitaria sanguinalis, and eleusine indica.	210	7
FLOODINGS SP	Classification of floodings events between 2015 and 2016 in São Paulo.	825	6

The attributes in each dataset vary in type and number. However, they all allow being tampered with generated noise, a necessary characteristic for evaluating the proposed methodology. All numerical values were tampered with multiplicative and additive noise.

It is interesting to note that all variables observed in real life are analog by nature, and we depend on instruments or sensors to measure them correctly [44]. In this sense, any data expressing some environmental characteristic will be expressed in real numerical values, making the numerical framework presented here perfectly applicable and coherent. It is impossible to apply the noises if the attributes were nominal or strings without probably changing their class, which would reflect in a dataset utterly different from the original.

3.2. WB Score Explained

Let $D_0 = \{(x_i, y_i)\ \epsilon\ X \times Y\ :\ i = 1, \ldots, m\}$ an original considered dataset; where x_i are observations and y_i a class label. Additionally, let $D_e = \{(x_i^{(e)}, y_i)\ \in\ X \times Y\ :\ i = 1, \ldots, m\}$ a dataset with observations tampered with e% noise-level.

Assuming $f_{\mathrm{k}}(x) : X \rightarrow Y$ a trained classifier through $\mathrm{D_k}$, for $k = 0, 1, \ldots, \beta$; the accuracy regarding the original and noise-corrupted datasets, herein denoted by $\mathrm{A_0}$ and $\mathrm{A_c}$, respectively, are defined as:

$$A_0 = \frac{1}{m} \sum_{i=1}^{m} \delta(f_{\mathrm{k}}(x_{\mathrm{i}}), y_{\mathrm{i}})$$

$$A_{\mathrm{c}} = \frac{1}{\beta.m} \sum_{k=1}^{\beta} \sum_{i=i}^{m} \delta(f_{\mathrm{k}}(x_{\mathrm{i}}), y_{\mathrm{i}})$$

where $\delta(a, b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ otherwise} \end{cases}$ is the Kronecker operator.

The basic idea of the proposed methodology is the analysis of possible models that may be used in a classification problem from a dataset that may contain noise in its data. The central evaluation metric used is the overall accuracy [45].

Applying different random noise intensities is a convenient procedure to assess the robustness of the model in this scenario. One can consider the original dataset as the baseline to be compared with other noise levels. Consequently, these perturbed datasets allow for analyzing how the error rates increase due to the added noise intensities.

For this discussion, we consider noise intensities ranging from ℓ_{inf}% to ℓ_{sup}%, by a progressive increase of ℓ_{step}%. Regarding the perturbation process, the following expression is employed:

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \frac{\epsilon \ell \mathbf{x}_i}{2};\ i = 1, 2, \ldots, m \tag{1}$$

where $\tilde{\mathbf{x}}_i$ is the resulting perturbation on $\mathbf{x}_i$; ϵ is a random scalar sampled from a uniform distribution in $[-1, +1]$; and $\ell \in [\ell_{inf}, \ell_{sup}]$ represents the noise level ranging from ℓ_{inf}% to ℓ_{sup}%, respectively.

It is worth observing that the respective assignment between $\mathbf{x}_i$ and its "label" ($\mathbf{y}_i$) is still persistent to $\tilde{\mathbf{x}}_i$, for $i = 1, \ldots, m$.

In addition to calculating the accuracy achieved by the model when using the original dataset ($\mathrm{A_0}$), the average of the accuracies of this same dataset with added noise ranging from ℓ_{inf}% to ℓ_{sup}%, with increment step of ℓ_{step}%, is also calculated ($\mathrm{A_c}$). Limiting the error or noise threshold in data measurement readings to a minimal or negligible value is one of the main goals when building a dataset, ensuring the assertiveness of the classification algorithm output.

Once the accuracy values are calculated, we can derive the ρ and θ values that comprise the indicators of the proposed methodology, herein called the WB Score. Figure 1 describes the above-discussed methodology in a flow chart diagram.

In practice, the accuracy values are placed on a Cartesian graph with the original accuracy on the y-axis and the average of the accuracies with noise on the x-axis, as depicted in Figure 2. The vector resulting from the sum of these values indicates the evaluation of the classification algorithm tested, where ρ is the value of its module and θ is the angle formed between the vector and the abscissa axis.

It is worth noting that, in this configuration, the ρ module can be greater than 1. In order to keep it between [0, 1], we can normalize it. In this way, the ρ and θ values can be expressed as follows:

$$\rho = \left(\frac{{A_0}^2 + {A_{\mathrm{c}}}^2}{2} \right)^{\frac{1}{2}};\quad [0, 1] \tag{2}$$

$$\theta = \arctan\left(\frac{A_0}{A_{\mathrm{c}}} \right);\quad (0, \pi/2). \tag{3}$$

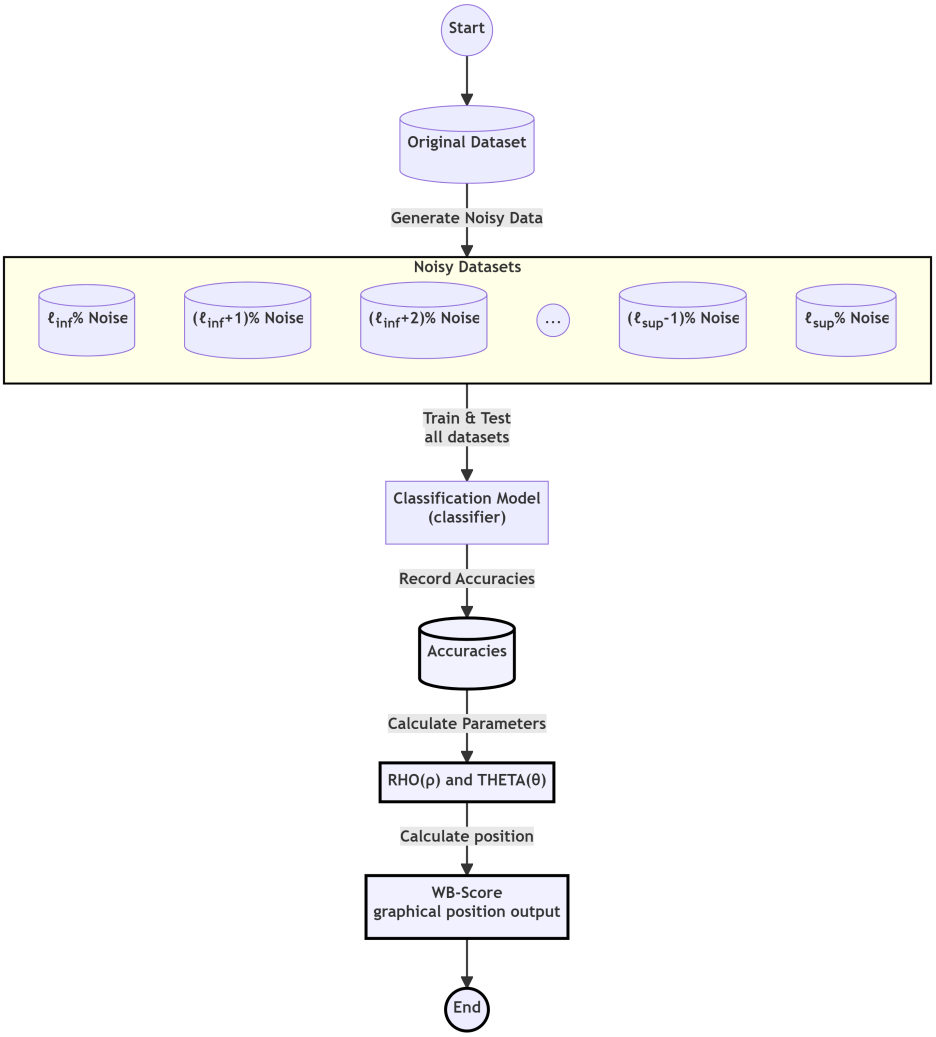

Figure 1. The flow chart shows how the WB Score methodology calculates the rho and theta indicators and the visual positioning of each classifier.

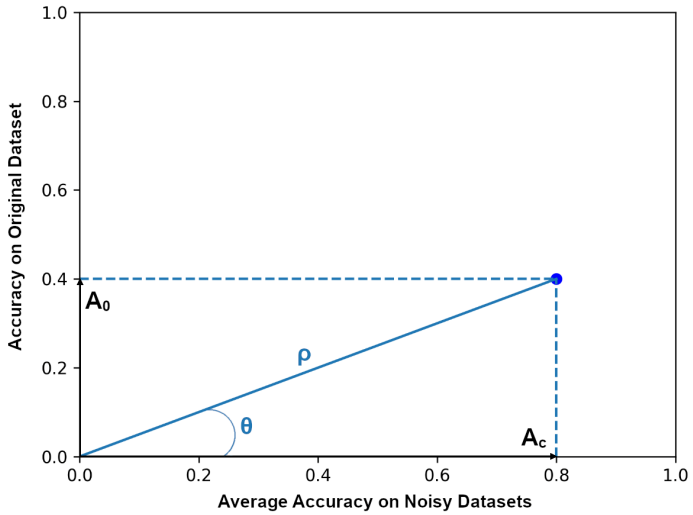

Figure 2. Visualization of ρ and θ indicators.

Analyzing Figure 2, we can point to the following properties:

(i) Vectors with $\theta < \frac{\pi}{4}$ generally represent algorithms with robustness in relation to noisy datasets;
(ii) Vectors with $\theta > \frac{\pi}{4}$ generally represent algorithms with robustness in relation to noiseless datasets;
(iii) Vectors with $\theta \approx \frac{\pi}{4}$ represent algorithms with a balanced response between noisy and noiseless datasets.

Figure 3 graphically express these properties, highlighting regions for each one of them. The region comprising the balanced response was arbitrarily defined to θ between 43 and 47 degrees (45 $\pm$ 2 degrees).

The indicative vectors may have the same modulus value (length of ρ) but can be differentiated by their angular part (value of θ). Thus, it is possible to choose a more robust algorithm for the case of a noisy dataset, a noiseless dataset, or a balanced solution between both.

The methodology will always return a position of the tested classifier within the delimited graphic area, except for the case in which the accuracies have a value of zero. In this case, we will have a mathematical uncertainty to calculate the arc whose tangent would be zero divided by zero. However, this is an undesirable situation, meaning that the classifier always gets its predictions wrong, and is therefore useless.

Lastly, it is worth mentioning that the accuracy evaluation metric is used in this introduction to the WB Score methodology. However, it can be changed to any other evaluation metric among dozens of others described in the literature, such as Kappa, Total Cost, Average Cost, KB, MAE, RMSE, RAE, RRSE Precision, Recall, F-measure, ROC, PRC, and others [46].

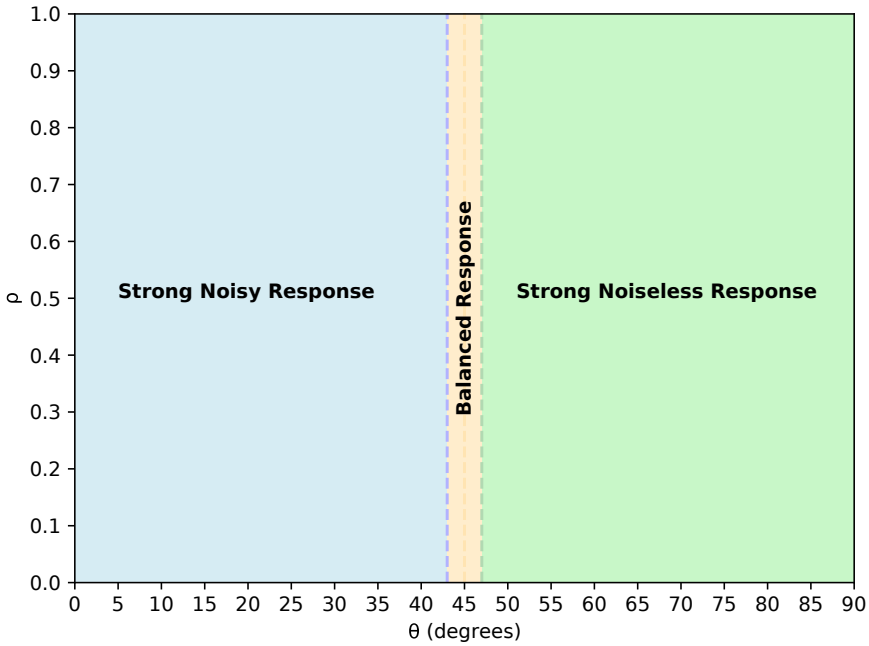

Figure 3. Regarding the values of the θ angle, we can identify distinct areas that indicate the robustness of the classification algorithm analyzed in terms of noisy, noiseless, or balanced response.

3.3. Selected Classifiers for Testing

Eight supervised classification algorithms were tested: KNN (K-Nearest Neighbors) [47]; Naïve Bayes (NB) [48]; Random Forest (RF) [49]; J48 (JAVA implementation of the original C4.5 classifier [50]); Random Tree (RT) [51]; Multilayer Perceptron (MLP) [52]; SVM with linear kernel [53]; and SVM with Radial Basis Function kernel [54]. One is a linear classifier (SVM with linear kernel), one implements the Bayes theory (Naïve Bayes), and all others are nonlinear-based classifiers. Table 2 lists all of them.

Table 2. List of selected classifiers for testing.

Classifier Name	Acronym	Classification Type
Naive Bayes	NB	Bayes Theory
Support Vector Machine (linear kernel)	SVM (linear)	Linear
C4.5 (ported in JAVA)	J48	Nonlinear
K-Nearest Neighbor	KNN	Nonlinear
MultilayerPerceptron	MLP	Nonlinear
Random Forest	RF	Nonlinear
Random Tree	RT	Nonlinear
Support Vector Machine (Radial Basis Function kernel)	SVM (RBF)	Nonlinear

3.4. The WEKA Software

WEKA (Waikato Environment for Knowledge Analysis) is a widely used and highly regarded open-source software suite for machine learning and data mining tasks. Developed at the University of Waikato in New Zealand, WEKA provides a comprehensive set of tools and algorithms for machine learning, including data preprocessing, classification, regression, clustering, association rules mining, and feature selection [55].

One of the primary strengths of WEKA lies in its extensive collection of classification algorithms. It offers a diverse range of techniques that can be employed to build predictive models from labeled datasets. These algorithms encompass traditional and state-of-the-art approaches, allowing researchers and practitioners to choose the most suitable method for their tasks. WEKA includes popular classification algorithms such as decision trees, random forests, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), naive Bayes, and neural networks.

WEKA's classification algorithms excel in their flexibility and configurability. The software provides users with numerous options to customize the learning process, such as adjusting parameters, handling missing values, nominal and numeric attributes, and imbalanced datasets. This level of control allows for fine-tuning the models and adapting them to different types of data and problem domains.

In this experiment, the WEKA exploration module was extensively used to follow the methodology described in Section 3.2. All the classifiers' implementations came from this module, included in the WEKA 3.9.6 version.

3.5. Fine Tuning with Grid Search

The Grid Search procedure [56] was used under the WEKA tool to fine-tune the classifiers' core parameters. Table 3 shows the key parameters each classifier was fine-tuned, the original search space entered, and the actual best configuration found. All parameter names are referenced as they are found within implementations of the classifiers in WEKA.

3.6. Introduced Noises

In this experiment, three types of noise were introduced to generate the noisy test datasets:

1. Multiplicative: random variations to the original value ensuring that the noise is centered around zero;
2. Additive: random variations to the original value with the variations centered around the mean value;
3. Both multiplicative and additive: noises are added and divided by two, so the result will remain within the desired noise level.

Table 3. Best parameters' configuration found using the Grid Search method.

Classifier	Fine-Tuned Parameter	Search Space	Best Configuration (IRIS, GLASS, IONOSPHERE, SEGMENTATION, SEEDS, FLOODINGS SP)
KNN	KNN distanceWeighting	{3, 5, 7, 9, 11} {None, 1/distance, 1-distance}	11, 3, 7, 3, 3, 3 1/distance (all)
NB	useKernelEstimator useSupervisedDiscretization	{0, 1} {0, 1}	1, 0, 1, 0, 0, 0 0, 1, 0, 1, 1, 1
RF	numIterations maxDepth	{10, 20, . . . , 190, 200} {1, 2, 3, . . . , 8, 9, 10}	200, 30, 60, 200, 80, 200 3, 7, 6, 10, 10, 7
J48	minNumObj unpruned	{1, 3, 5, 7, 9, 11} {0, 1}	3, 5, 5, 5, 1, 1 1 (all)
RT	breakTiesRandomly maxDepth	{0, 1} {1, 2, 3, . . . , 8, 9, 10}	1, 1, 0, 1, 0, 0 2, 6, 6, 5, 5, 9
MLP	learningRate momentum	{0.1, 0.2, . . . , 0.5} {0.1, 0.2, . . . , 0.5}	0.5, 0.5, 0.4, 0.5, 0.5, 0.5 0.5, 0.2, 0.1, 0.1, 0.1, 0.4
SVM (linear)	cost coef0	{1, 10, 100, 1000, 10000} {0, 1}	10000, 10, 100, 1, 100, 100 1 (all)
SVM (RBF)	cost gamma	{1, 10, 100, 1000, 10000} {0.01, 0.1, 1}	10, 10, 10, 10000, 10000, 1 0.01, 1, 0.1, 10000, 0.01, 0.1

These types of noises appear when an output response depends on a linear combination of input data, the typical scenario in production environments. A linear combination involves the accumulation of additive and multiplicative errors, which is the behavior reproduced by this approach.

For the generation of each test dataset with a specified noise level, the original dataset was read line by line, and all numerical values were tampered with these three noise types, generating 100 new instances each. Only the label value remained the same. Ultimately, each test dataset was 300 times bigger than the original one.

More information about this procedure, the code, and the public datasets used can be found in the GitHub repository from this experiment: https://github.com/wagnerbilla/WBScore (accessed on 1 July 2023).

3.7. Accuracies Calculation

For the application of the methodology described in Section 3.2, the following values were used: $\ell_{inf} = 1$, $\ell_{sup} = 20$ and $\ell_{step} = 1$. Noises of progressive intensities from 1 to 20% were considered, with an incremental step of 1%. It was supposed that noise levels above 20% are no longer a concrete basis for comparing with the original dataset.

For each of the datasets described in Section 3.1 and for each classifier described in Section 3.3, the following steps were applied to calculate the accuracies:

(i) The classifier was trained on the original dataset;
(ii) The classifier was tested on the original dataset;
(iii) For each of the 20 datasets with different noise levels, the classifier was tested, and the accuracies were registered;
(iv) A graph of the classifier as a function of the noise level was plotted.

The graph plots showed that the accuracy of the classifier generally decreases as the noise level increases.

The described approach is quick and direct for measuring accuracy values as a function of introduced noise. The intention was not to guarantee the absence of data overfits by the

classifiers; it is a reference implementation only. One can intend to replace it completely, using any other technique found in literature, splitting the original dataset into three others (train, validation, and test), and applying noise only in the tested instances or including a cross-validation scheme during the training phase.

4. Results and Discussion

4.1. Classic Datasets

Figure 4 shows the accuracy curves of each classifier tested over the variation of the inserted noise, using the best parameters' configurations found with the Grid Search method in all selected testing datasets, as detailed earlier in Table 3.

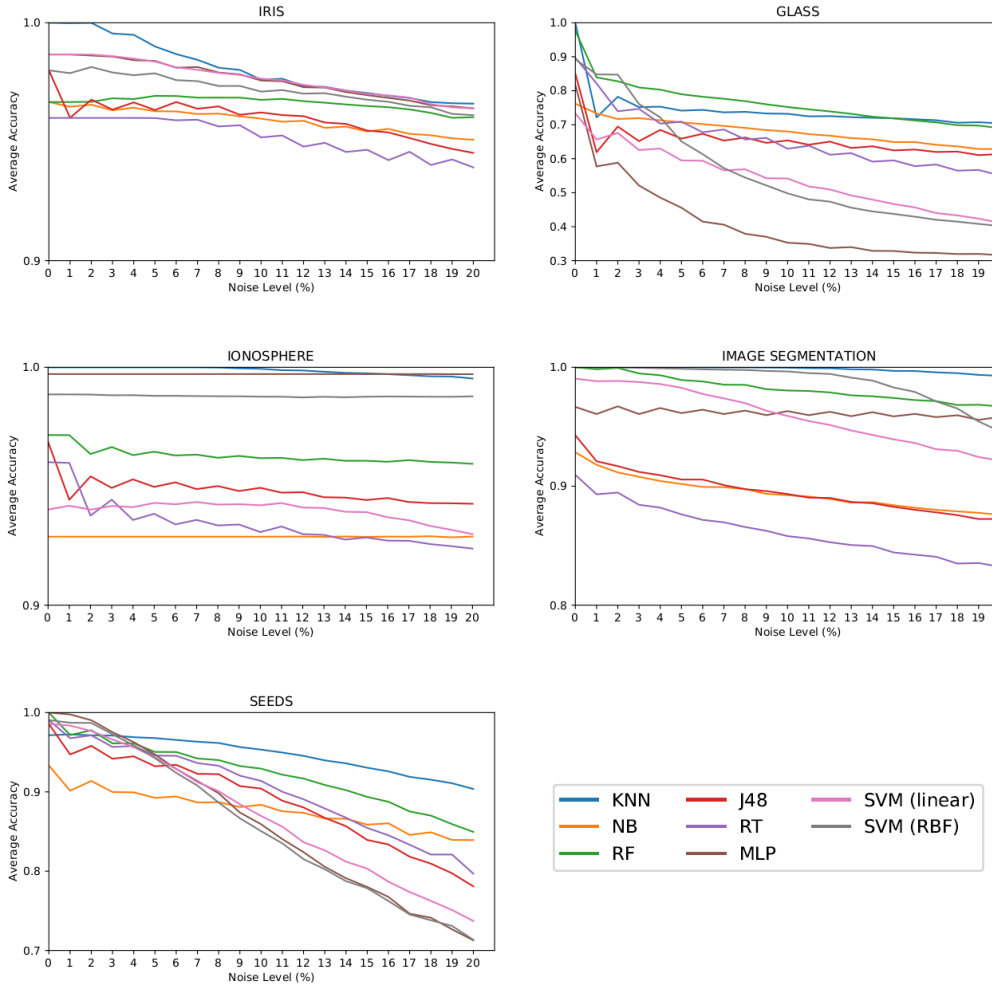

Figure 4. The tested algorithms' performance curves as a function of adding noise levels to the original datasets.

In all graphs, a generalized degradation of the accuracy of the classifiers is visible due to the increase in noise. The response curves of the classifiers also change concerning the dataset type because it is a data-driven process, and each dataset type generates a different "signature" by each classifier.

The classifiers that stood out in these tests were the KNN and the RF. KNN won in the IRIS, IONOSPHERE, IMAGE SEGMENTATION, and SEEDS datasets, whereas in the GLASS dataset, the RF gives the best response. They all have better overall accuracy than their competitors, almost always remaining as the highest curves in the performance graphs.

Figure 5 shows the position of each classifier in the WB Score graph space for each tested dataset.

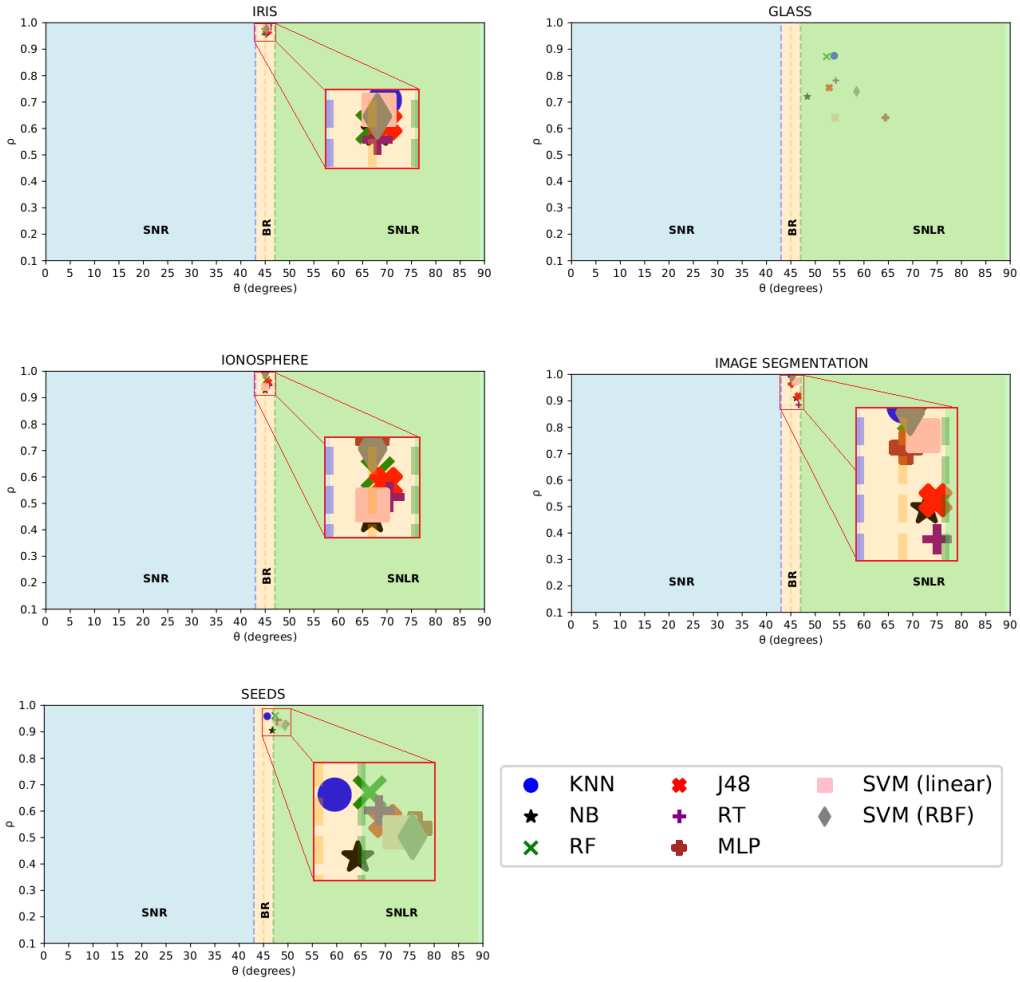

Figure 5. The classifier's position in the WB Score graph space according to ρ and θ values.

Each graph represents a visual interpretation of all classifiers as a function of the tested dataset and the chosen evaluation metric, which, in this case, is accuracy. It is possible to perceive a greater or lesser spread of the position of the classifiers in each of the results, which may indicate the internal structure of the dataset.

We could analyze all of these graphical results, but for brevity, let us look closer at the SEEDS dataset, as the spread of the classifiers is concentrated on two regions of the graph that can be explored: Balanced Response (BR) and Strong Noiseless Response (SNLR). Remember that what is discussed here for this dataset is also valid for the others. Table 4 shows the summarized results and Table 5 the classifiers' performance rank for the SEEDS dataset.

Table 4. Summarized results of WSB Score's parameters for the SEEDS dataset.

Algorithm	A_0	A_c	ρ	θ	Accuracy Drop %
KNN	0.971	0.946	0.958	45.74	6.78
NB	0.933	0.875	0.904	46.82	9.40
RF	1	0.920	0.960	47.38	15.05
J48	0.985	0.884	0.936	48.10	20.49
RT	0.990	0.897	0.945	47.80	19.36
MLP	1	0.854	0.929	49.49	28.72
SVM (linear)	0.985	0.863	0.926	48.78	24.85
SVM (RBF)	0.990	0.849	0.922	49.37	27.70

Table 5. Performance rank of the eight tested classifiers against the SEEDS dataset.

Rank #	Overall Performance (ρ)	Strong Noiseless Response ($\rho \sin \theta$)	Balanced Response ($43° \leq \theta \leq 47°$)
1	RF	RF	KNN
2	KNN	MLP	NB
3	RT	RT	
4	J48	SVM (RBF)	
5	MLP	J48	
6	SVM (linear)	SVM (linear)	
7	SVM (RBF)	KNN	
8	NB	NB	

For the SEEDS dataset, we have the ranking of the best classifiers for three possible situations: overall performance (when considering only the ρ value), strong noiseless response (an environment without noise), and a balanced response. Thus, the practitioner can decide which is the best classifier for his dataset in the environment in which it should be inserted.

Finally, it is also worth noting that the KNN classifier, in addition to presenting the best accuracy, also had the lowest drop in accuracy of all classifiers tested, making it a great choice for classifying this dataset. This information can be seen in the "Accuracy drop %" column in Table 4.

4.2. Customized Flooding Dataset

Figure 6 shows the accuracy curves of each classifier tested against a custom flooding dataset using the WB Score methodology. These data came from a real production environment and were taken from the Climate Emergency Management Center (CGE) between 2015 and 2016 (https://www.cgesp.org/). Additional information on this dataset can be found in Section 2.1 of [43].

Again we have different responses for each type of classifier, showing that the internal and logical implementation of each has different approaches. The SVM (RBF), SVM (linear), and MLP classifiers practically behaved indifferently to the applied noise level, always presenting similar accuracy throughout the process. This situation did not occur in any of the five classic datasets analyzed earlier, indicating that this behavior is linked to the data contained in the dataset and how its random variables behave. The other classifiers had their outputs degraded, as expected. KNN was the winning classifier, followed by RF.

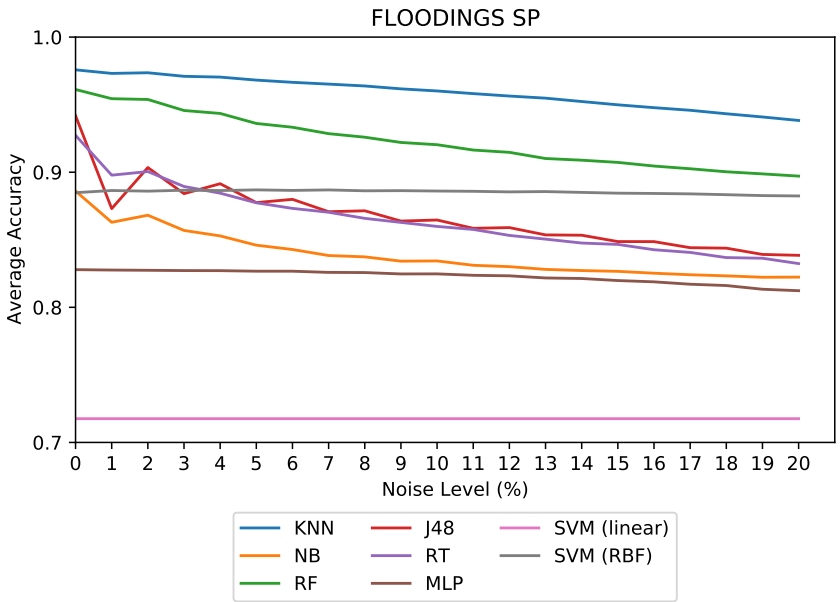

Figure 6. The tested algorithms' performance against the FLOODINGS SP dataset.

Figure 7 shows the position of each classifier in the WB Score graph space for the FLOODINGS SP dataset.

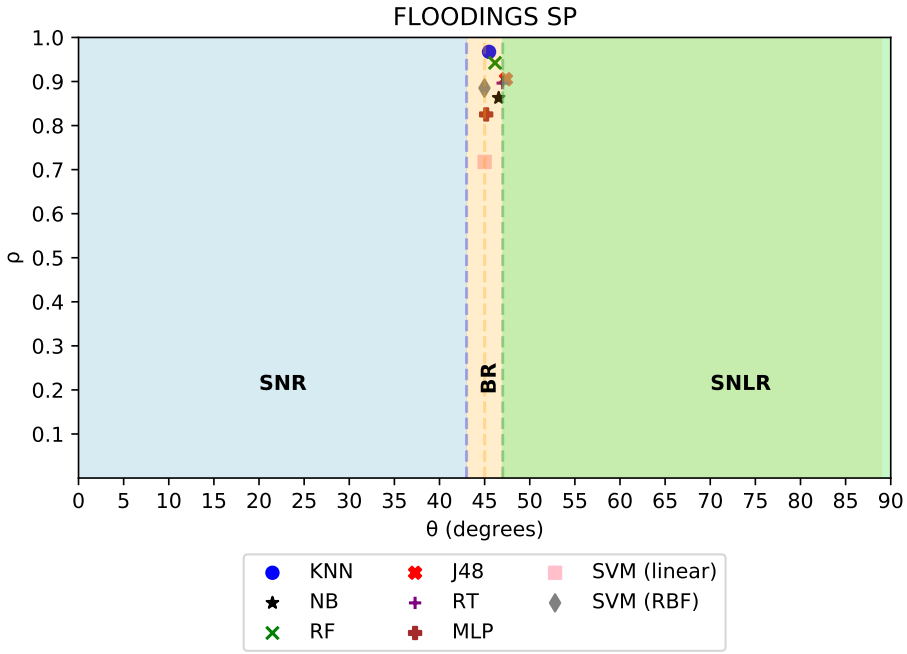

Figure 7. The classifier's position in the WB Score graph space for the FLOODINGS SP dataset.

It is possible to verify that almost all classifiers are located in the Balanced Response (BR) area; only the classifiers based on decision trees RT and J48 are located in the Strong Noiseless Response Area (SNRA). At the top of the graph, we can easily see the icons corresponding to the KNN and RF algorithms.

Table 6 shows the summarized results and Table 7 the classifiers' performance rank for the FLOODING SP dataset.

Table 6. Summarized results of WSB Score's parameters for the FLOODINGS SP dataset.

Algorithm	A_0	A_c	ρ	θ	Accuracy Drop %
KNN	0.975	0.958	0.967	45.49	3.74
NB	0.886	0.839	0.862	46.56	6.37
RF	0.961	0.923	0.942	46.15	6.41
J48	0.941	0.867	0.905	47.36	10.33
RT	0.927	0.864	0.896	47.00	9.49
MLP	0.827	0.822	0.825	45.17	1.55
SVM (linear)	0.717	0.717	0.717	45.00	0
SVM (RBF)	0.884	0.885	0.885	44.98	0

Table 7. Performance rank of the eight tested classifiers against the FLOODINGS SP dataset.

Rank #	Overall Performance (ρ)	Strong Noiseless Response ($\rho \sin \theta$)	Balanced Response $43° \leq \theta \leq 47°$)
1	KNN	KNN	SVM (RBF)
2	RF	RF	SVM (linear)
3	J48	J48	MLP
4	RT	RT	KNN
5	SVM (RBF)	NB	RF
6	NB	SVM (RBF)	NB
7	MLP	MLP	RJ
8	SVM (linear)	SVM (linear)	J48

According to the results presented in the tables above, the nonlinear classifiers KNN and RF presented the best results for the FLOODING SP dataset, with KNN being the best option. It is interesting to note that the SVM's classifiers presented no accuracy drop in all noise levels tested, indicating that they are immune to the noise applied to this dataset.

4.3. Comparison of Classifier Selection Methods

The proposed WB Score methodology addresses the challenges of selecting robust classifiers in noisy real-world scenarios. To provide a comprehensive comparison, we will evaluate the WB Score methodology alongside other standard methods: ensemble learning, Grid Search with cross-validation, performance-based selection, algorithm selection heuristics, algorithm ranking based on statistical tests, and portfolio selection.

WB Score Methodology: The WB Score methodology stands out for explicitly addressing noise-related challenges. It focuses on classifier performance and noise robustness, which is crucial in real-world applications. Utilizing a visually intuitive graph for performance representation aids decision-making.

Ensemble Learning: Ensemble learning combines predictions to improve performance. It enhances robustness by reducing noise impact. However, it may increase computational complexity and needs explicit insights into individual classifier performance [57,58].

Grid Search with Cross-Validation: Grid Search with cross-validation systematically explores hyperparameter combinations for classifiers. It helps find optimal configurations but can be computationally expensive [59].

Performance-Based Selection: Performance-based selection trains classifiers on a subset of data and evaluates their performance on a validation set. It prioritizes the best-performing classifiers, assuming consistent noise levels between training and validation data [60].

Algorithm Selection Heuristics: Algorithm selection heuristics use dataset characteristics to guide classifier choice. Although they can be effective, they may not account for subtle noise patterns [61].

Algorithm Ranking Based on Statistical Tests: This method ranks classifiers based on their performance across multiple data splits and uses statistical tests to determine significance. However, it may not fully capture noise-related challenges [62].

Portfolio Selection: Portfolio selection treats classifier selection as an optimization problem, distributing resources based on historical classifier performance. It requires careful consideration of noise and performance history [3].

Table 8 summarizes the method's main characteristics, highlighting each approach's pros and cons.

The WB Score methodology offers a specialized approach explicitly designed to address noise-related challenges, making it a promising solution for real-world scenarios with increasing noise levels. Although other methods, such as ensemble learning, Grid Search with cross-validation, performance-based selection, algorithm selection heuristics, algorithm ranking based on statistical tests, and portfolio selection, contribute valuable techniques, their application may require careful consideration of specific noise-related concerns.

Table 8. Comparison of classifier selection methods.

Method	Pros	Cons
WB Score Methodology	Explicitly addresses noise-related challenges.	Specific details of noise robustness and efficient handling are not detailed.
	Emphasizes robustness to noise and efficient noise handling.	
	Utilizes a visually intuitive graph for performance representation.	
Ensemble Learning	Robustness through combination of multiple classifiers.	Increased computational complexity.
	Effective noise reduction.	Limited insight into individual classifier performance.
Grid Search with Cross-Validation	Systematic exploration of hyperparameter space.	Computationally expensive for large search spaces.
	Can find optimal configurations.	
Performance-Based Selection	Prioritizes best-performing classifiers.	Assumes consistent noise levels between training and validation data.
	Evaluates classifiers on validation set.	
Algorithm Selection Heuristics	Guided by dataset characteristics.	May not capture subtle noise patterns.
	Tailored to specific characteristics.	
Algorithm Ranking Based on Statistical Tests	Ranks classifiers based on performance.	May not fully capture noise-related challenges.
	Uses statistical tests for significance.	
Portfolio Selection	Optimizes resource allocation based on historical performance.	Requires consideration of noise and performance history.
	Addresses classifier performance over time.	

5. Conclusions

According to the type of dataset used for a classification task, there is a better algorithm for the problem to be attacked in which the bias is minimized. In the vast majority of data, noise insertions make it challenging to choose the best robust single solution for the purpose, and there needs to be more literature on this type of approach. The WB Score methodology presented here makes it possible to choose the best classifier for a given dataset among a set of options, considering its accuracy response when introducing noise levels representing a scenario where classifiers are trained once and deployed in production environments, continuously facing increasing noise levels over time. Graphical or visual methodologies are found in the literature. However, none were designed to consider the increase in noise levels nor to analyze a set of classifiers simultaneously to select the best one.

Five classic datasets and a customized one were used to show that the methodology will always present its graphical output as long as the accuracy of the tested classifiers is nonzero and the dataset is based on numerical values. Two classifiers stood out in the tests, drawing much attention: KNN and RF, both considered nonlinear classifiers.

In real-life applications, the analyzed objects are usually represented as random variables. Contaminating it with additive and multiplicative noise is a convenient process to produce scenarios with distinct variability. In this sense, generalizing the results for tens of datasets in a noisy environment involves positioning the tested classifiers in the areas defined in the WB Score output graph. This straightforward procedure enables users to select the best classifier for their specific needs.

Possible WB Score methodology application fields are medical settings where accurate and reliable classification of patient data is paramount. The ability to explicitly account for noise and ensure robustness can lead to improved diagnostic accuracy; financial forecasting, where markets are noisy and subject to unpredictable fluctuations, the methodology's focus on noise robustness empowers financial analysts to identify classifiers better suited for handling noisy market data, leading to more accurate predictions and informed investment strategies; environmental monitoring, where the choice of the best classifier aids in building reliable models for predicting natural events, such as rainfall patterns or air quality levels, contributing to effective environmental management and disaster preparedness; and industrial settings, where data from sensors and machinery may be contaminated with noise, the WB Score methodology can optimize predictive maintenance processes by selecting robust classifiers that identify potential equipment failures and maintenance needs more accurately, minimizing downtime and maximizing operational efficiency.

This article introduced a basic reference implementation to present an overview of the WB Score methodology. This implementation can be changed according to the needs of practitioners. One can change, for example, how the training and testing phases of the classifiers are performed, as well as choose another evaluation metric.

6. Future Works

Well-known classic datasets and a custom dataset were explored here. However, there are a multitude of types with different characteristics among them. We can mention new scenarios to be studied: unbalanced datasets, noise in classes not yet seen, and the addition of between-class samples.

The types of noise introduced in this study were the traditional ones found in the literature, especially when the classifier's response is a function of a linear combination of random variables in the dataset's attributes. However, other types should be explored, such as the generation of convolutional noise and the reinforcement method commonly used in reinforcement learning scenarios.

Author Contributions: Conceptualization, methodology, software, and analysis: W.S.B.; writing: W.S.B., R.G.N. and L.B.L.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the São Paulo Research Foundation (FAPESP), grant 2021/01305-6, and National Council for Scientific and Technological Development (CNPq), grant 305220/2022-5.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://archive.ics.uci.edu/ml/datasets (accessed on 1 July 2023). All relevant data produced by the authors can also be found at: https://github.com/wagnerbilla/WBScore.

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

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
2. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson: London, UK, 2019.
3. Caruana, R.; Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2006; pp. 161–168.
4. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: Berlin/Heidelberg, Germany, 2009.
5. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 2096–2130.
6. Barr, T.A.; Neyshabur, B. Revisiting small batch training for deep neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4320–4330.
7. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.

8. Datta, A.; Sen, S.; Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In Proceedings of the 37th IEEE Symposium on Security and Privacy, San Jose, CA, USA, 23–25 May 2016; pp. 598–617.
9. Reed, S.E.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 1462–1471.
10. Camps-Valls, G.; Bruzzone, L. Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.* **2009**, *47*, 932–945. [CrossRef]
11. Sievert, C.; Shirley, K. LDAvis: A method for visualizing and interpreting topics. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2018; pp. 564–574.
12. Seabold, S.; Perktold, J. Statsmodels: Econometric and statistical modeling with Python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; pp. 92–96.
13. Sen, P.; Hajra, M.; Ghosh, M. Supervised classification algorithms in machine learning: A survey and review. In *Emerging Technology in Modelling and Graphics*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 99–111.
14. Swan, A.; Mobasheri, A.; Allaway, D.; Liddell, S.; Bacardit, J. Application of machine learning to proteomics data: Classification and biomarker identification in postgenomics biology. *Omics J. Integr. Biol.* **2013**, *17*, 595–610. [CrossRef]
15. Kung, S.; Mak, M.; Lin, S.; Mak, M.; Lin, S. *Biometric Authentication: A Machine Learning Approach*; Prentice Hall Professional Technical Reference: New York, NY, USA, 2005.
16. Brunetti, A.; Buongiorno, D.; Trotta, G.; Bevilacqua, V. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing* **2018**, *300*, 17–33. [CrossRef]
17. Khan, A.; Baharudin, B.; Lee, L.; Khan, K. A review of machine learning algorithms for text-documents classification. *J. Adv. Inf. Technol.* **2010**, *1*, 4–20.
18. Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; et al. Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* **2019**, *18*, 463–477. [CrossRef] [PubMed]
19. Sharma, P.; Kaur, M. Classification in pattern recognition: A review. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2013**, *3*, 1–9.
20. Kumar, A.; Irsoy, O.; Ondruska, P.; Iyyer, M.; Bradbury, J.; Gulrajani, I.; Zhong, V.; Paulus, R.; Socher, R. Ask me anything: Dynamic memory networks for natural language processing. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1378–1387.
21. Adam, S.; Alexandropoulos, S.; Pardalos, P.; Vrahatis, M. No free lunch theorem: A review. In *Approximation and Optimization*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 57–82.
22. Gómez, D.; Rojas, A. An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification. *Neural Comput.* **2016**, *28*, 216–228. [CrossRef] [PubMed]
23. Khan, I.; Zhang, X.; Rehman, M.; Ali, R. A literature survey and empirical study of meta-learning for classifier selection. *IEEE Access* **2020**, *8*, 10262–10281. [CrossRef]
24. Brazdil, P.; Soares, C. A comparison of ranking methods for classification algorithm selection. In Proceedings of the European Conference on Machine Learning, Barcelona, Spain, 31 May–2 June 2000; pp. 63–75.
25. Soares, C.; Brazdil, P. Zoomed ranking: Selection of classification algorithms based on relevant performance information. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Lyon, France, 13–16 September 2000; pp. 126–135.
26. Pacheco, A.; Krohling, R. Ranking of classification algorithms in terms of mean–standard deviation using A-TOPSIS. *Ann. Data Sci.* **2018**, *5*, 93–110. [CrossRef]
27. Abdulrahman, S.; Brazdil, P.; Zainon, W.; Adamu, A. Simplifying the algorithm selection using reduction of rankings of classification algorithms. In Proceedings of the 2019 8th International Conference on Software and Computer Applications, Penang, Malaysia, 19–21 February 2019; pp. 140–148.
28. Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J.; Larochelle, H.; Zemel, R. Meta-learning for semi-supervised few-shot classification. *arXiv* **2018**, arXiv:1803.00676.
29. Wu, J.; Xiong, W.; Wang, W. Learning to learn and predict: A meta-learning approach for multi-label classification. *arXiv* **2019**, arXiv:1909.04176.
30. Brun, A.; Britto, A., Jr.; Oliveira, L.; Enembreck, F.; Sabourin, R. A framework for dynamic classifier selection oriented by the classification problem difficulty. *Pattern Recognit.* **2018**, *76*, 175–190. [CrossRef]
31. Kalousis, A.; Theoharis, T. Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intell. Data Anal.* **1999**, *3*, 319–337.
32. Cruz, R.; Sabourin, R.; Cavalcanti, G. Dynamic classifier selection: Recent advances and perspectives. *Inf. Fusion* **2018**, *41*, 195–216. [CrossRef]
33. Hasan, R.; Chu, C. Noise in Datasets: What Are the Impacts on Classification Performance? In Proceedings of the ICPRAM, Online, 3–5 February 2022; pp. 163–170.
34. Saseendran, A.; Setia, L.; Chhabria, V.; Chakraborty, D.; Barman Roy, A. Impact of noise in dataset on machine learning algorithms. *Mach. Learn. Res.* **2019**, *1*, 1–8.

35. Xiao, J.; He, C.; Jiang, X.; Liu, D. A dynamic classifier ensemble selection approach for noise data. *Inf. Sci.* **2010**, *180*, 3402–3421. [CrossRef]
36. Zhu, X.; Wu, X.; Yang, Y. Dynamic classifier selection for effective mining from noisy data streams. In Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04), Brighton, UK, 1–4 November 2004; pp. 305–312.
37. Krawczyk, B.; Cano, A. Online ensemble learning with abstaining classifiers for drifting and noisy data streams. *Appl. Soft Comput.* **2018**, *68*, 677–692. [CrossRef]
38. Pichery, C. Sensitivity analysis. In *Encyclopedia of Toxicology*, 3rd ed.; Elsevier: Amsterdam, The Netherlands, 2014.
39. Oliveira Simoyama, F.; Tomás, L.; Pinto, F.; Salles-Neto, L.; Santos, L. Optimal rain gauge network to reduce rainfall impacts on urban mobility—A spatial sensitivity analysis. *Ind. Manag. Data Syst.* **2022**, *122*, 2261–2280. [CrossRef]
40. Saltelli, A.; Tarantola, S.; Campolongo, F.; Ratto, M. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*; Wiley Online Library: Hoboken, NJ, USA, 2004.
41. Morris, M. Factorial sampling plans for preliminary computational experiments. *Technometrics* **1991**, *33*, 161–174. [CrossRef]
42. Sin, G.; Gernaey, K. Improving the Morris method for sensitivity analysis by scaling the elementary effects. *Comput. Aided Chem. Eng.* **2009**, *26*, 925–930.
43. Silva Billa, W.; Santos, L.; Negri, R. From rainfall data to a two-dimensional data-space separation for flood occurrence. *An. Do(A) Encontro Nac. Model. Comput. Encontro Ciênc. Tecnol. Mater.* **2021**. . [CrossRef]
44. Kerr, A.R.; Randa, J. Thermal Noise and Noise Measurements—A 2010 Update. *IEEE Microw. Mag.* **2010**, *11*, 40–52. [CrossRef]
45. Hossin, M.; Sulaiman, M. A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manag. Process* **2015**, *5*, 1–11.
46. Powers, D. Evaluation: From precision, recall and f-measure to roc, auc and informedness. *J. Mach. Learn. Res.* **2011**, *12*, 2137–2163.
47. Kramer, O.; Kramer, O. K-nearest neighbors. In *Dimensionality Reduction with Unsupervised Nearest Neighbors*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 13–23.
48. Rish, I.; Smith, J.; Johnson, A.; Davis, M. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, WA, USA, 4–6 August 2001; Volume 3, pp. 41–46.
49. Cutler, A.; Cutler, D.; Stevens, J. Random forests. *Ensemble Mach. Learn.* **2012**, *45*, 157–175.
50. Ruggieri, S. Efficient C4. 5 [classification algorithm]. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 438–444. [CrossRef]
51. Le Gall, J. Random trees and applications. *Probab. Surv.* **2005**, *2*, 245–311. [CrossRef]
52. Murtagh, F. Multilayer perceptrons for classification and regression. *Neurocomputing* **1991**, *2*, 183–197. [CrossRef]
53. Bhavsar, H.; Panchal, M. A review on support vector machine for data classification. *Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET)* **2012**, *1*, 185–189.
54. Liu, Q.; Chen, C.; Zhang, Y.; Hu, Z. Feature selection for support vector machines with RBF kernel. *Artif. Intell. Rev.* **2011**, *36*, 99–115. [CrossRef]
55. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18. [CrossRef]
56. Syarif, I.; Prugel-Bennett, A.; Wills, G. SVM parameter optimization using grid search and genetic algorithm to improve classification performance. *TELKOMNIKA Telecommun. Comput. Electron. Control* **2016**, *14*, 1502–1509. [CrossRef]
57. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
58. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
59. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
60. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2015**, *27*, 1–30. [CrossRef]
61. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181.
62. Pareja, J.A.; Weber, R. Statistical comparison of classifiers through a cross-fitting approach. *Pattern Recognit. Lett.* **2014**, *36*, 105–112.