Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2020, 8, 341-357
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Basic Tenets of Classification Algorithms *K*-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review

Ernest Yeboah Boateng[1], Joseph Otoo[2], Daniel A. Abaye[1*]

[1]Department of Basic Sciences, School of Basic and Biomedical Sciences, University of Health and Allied Sciences, Ho, Ghana
[2]Department of Statistics and Actuarial Science, University of Ghana, Accra, Ghana
Email: eyboateng@uhas.edu.gh, jotoo011@st.ug.edu.gh, *dabaye@uhas.edu.gh

How to cite this paper: Boateng, E.Y., Otoo, J. and Abaye, D.A. (2020) Basic Tenets of Classification Algorithms *K*-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. *Journal of Data Analysis and Information Processing*, **8**, 341-357.
https://doi.org/10.4236/jdaip.2020.84020

Received: July 13, 2020
Accepted: November 20, 2020
Published: November 23, 2020

Abstract

In this paper, sixty-eight research articles published between 2000 and 2017 as well as textbooks which employed four classification algorithms: *K*-Nearest-Neighbor (KNN), Support Vector Machines (SVM), Random Forest (RF) and Neural Network (NN) as the main statistical tools were reviewed. The aim was to examine and compare these nonparametric classification methods on the following attributes: robustness to training data, sensitivity to changes, data fitting, stability, ability to handle large data sizes, sensitivity to noise, time invested in parameter tuning, and accuracy. The performances, strengths and shortcomings of each of the algorithms were examined, and finally, a conclusion was arrived at on which one has higher performance. It was evident from the literature reviewed that RF is too sensitive to small changes in the training dataset and is occasionally unstable and tends to overfit in the model. KNN is easy to implement and understand but has a major drawback of becoming significantly slow as the size of the data in use grows, while the ideal value of *K* for the KNN classifier is difficult to set. SVM and RF are insensitive to noise or overtraining, which shows their ability in dealing with unbalanced data. Larger input datasets will lengthen classification times for NN and KNN more than for SVM and RF. Among these nonparametric classification methods, NN has the potential to become a more widely used classification algorithm, but because of their time-consuming parameter tuning procedure, high level of complexity in computational processing, the numerous types of NN architectures to choose from and the high number of algorithms used for training, most researchers recommend SVM and RF as easier and wieldy used methods which repeatedly achieve results with high accuracies and are often faster to implement.

Keywords

Classification Algorithms, Non-Parametric, *K*-Nearest-Neighbor, Neural Networks, Random Forest, Support Vector Machines

1. Introduction

In the last few decades, a large number of methods for classification have been developed [1]. Among the most widely used techniques are *K*-Nearest-Neighbors (KNN) [2], artificial neural networks (ANNs) [3] [4] [5] [6], support vector machines (SVMs) [7] [8] [9] and ensembles of classification trees such as random forest (RF) [10] [11].

Algorithms based on decision trees (DT), are easy to apply, as a fewer number of parameters need to be estimated; hence, these have high degrees of automation [12]. However, this comparative advantage of DT with respect to ANN can be hidden by a tendency to overfit data [13]. For these reasons, both ANN and DT are, in recent years, being replaced by more advanced, simpler to train machine learning algorithms (MLAs). During the past decade, the family of kernel methods such as SVM [14] [15] and ensembles of trees such as RF [16] [17] have emerged as very promising methodologies for classification purposes.

Several studies demonstrate that, MLAs are more accurate than statistical techniques such as discriminant analysis or logistic regression, especially when the feature space is complex or the input datasets are expected to have different statistical distributions [4] [9]. As computational power has increased, MLAs have gained greater attention and the quality of pattern recognition systems has also increased correspondingly [18]. Thus, in most classification studies, RF, KNN and SVM are reported as the foremost classifiers producing high accuracies [19].

The basic steps to decide which algorithm to use will depend on a number of factors such as the number of examples in training set, dimensions of featured space, whether there are correlated features and whether overfitting is a problem [20]. Once these concerns have been addressed, the algorithm to use is then decided. Using methods of statistical physics, the generalization performance of SVMs, which have been recently introduced as a general alternative to neural networks (NN), were investigated [21]. It was evident from the study that for nonlinear classification rules, the generalization error saturates on a plateau when the number of examples is too small to properly estimate the coefficients of the nonlinear part. When trained on simple rules, it was found that SVMs overfit only weakly [21]. The performance of SVMs is strongly enhanced when the distribution of the inputs has a gap in feature space.

To avoid human introduced biases, Raczko and Zagajewski [22] used a 0.632 bootstrap procedure to evaluate three nonparametric classification algorithms (SVM, RF and ANN) in an attempt to classify the five most common tree species. The classification results indicated that, ANN achieved the highest median overall classification accuracy (77%) followed by SVM with 68% and RF with

62%. Analysis of the stability of results concluded that RF and SVM had the lowest variance of overall accuracy and κ (*kappa*) coefficient (12 percentage points) while ANN had 15 percentage points variance in results. A study showed that there exist some data distributions where maximal unpruned trees used in the RF do not achieve as good performance as the trees with smaller number of splits and/or smaller node size [23]. This was an improvement on the work reported earlier that RF do not overfit as the number of trees grows [10]. Thus, application of RF in general requires careful tuning of the relevant classifier parameters [24]. Bosch *et al.* [25] demonstrated that using random forests/ferns with an appropriate node test reduces training and testing costs significantly over a multi-way SVM and has comparable performance.

The performances of various classification methods however, still depend greatly on the general characteristics of the data to be classified [26]. The exact relationship between the data to be classified and the performance of various classification methods still remains to be determined. Thus far, there has been no classification method that works best on any given problem [26]. There have been various problems associated with classification methods in current use [20]. Therefore, to determine the best classification method for a certain dataset, a trial and error approach is used to decide on the best performance.

In this review paper, the performances, strengths and shortcomings of the KNN, SVM, RF and NN classifiers are examined and compared. Answers to the following questions are sought. What are the strengths and weaknesses of these algorithms on a set of classification problems? Which one performs better and under what conditions does one classifier perform better than the others? The four nonparametric classification methods were therefore, evaluated on the following; robustness to training data, sensitivity to changes, data fitting, stability, ability to handle large data sizes, sensitivity to noise, time invested in parameter tuning, and accuracy.

2. Materials and Methods

2.1. Support Vector Machines (SVM)

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The SVM algorithm developed by Cortes and Vapnik [8] tries to find the optimal hyperplane in n-dimensional classification space with the highest margin between classes (**Figure 1**).

The SVM algorithm is often reported to achieve better results than other classifiers [9], although it has been indicated that the main reason to use an SVM instead is because the problem might not be linearly separable [27]. In that case, an SVM with a non-linear kernel such as the Radial Basis Function (RBF) would be suitable. Another related reason to use SVMs, is if one is in a high dimensional space. For example, SVMs have been reported to work better for text classification although this requires a lot of time for training [28].

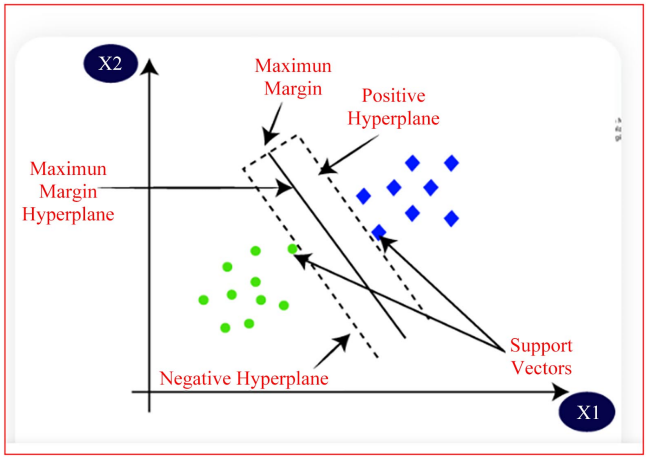

Figure 1. A simple illustration of the Support Vector Machine (SVM) algorithm in 2-dimensions.

The SVM is an extension of the support vector classifier and is obtained as a result from the enlargement of the feature space in a specific way, using kernels [29].

Representation of linear support vector classifier is as shown in Equation (1):

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle \tag{1}$$

where $\alpha_1, \cdots, \alpha_n$ and β_0 are parameters which are estimated by $\binom{n}{2}$ inner products $\langle x_i, x_i' \rangle$ between all pairs of training observations. Replacing the inner product with $K(x_i, x_i')$, where K is some function called the kernel. Linear kernel is represented as shown in Equation (2):

$$K(x_i, x_i') = \sum_{j=1}^{p} x_{ij} x_{ij}' \tag{2}$$

Polynomial kernel of degree d (where d is positive) can be represented as shown in Equation (3):

$$K(x_i, x_i') = \left(1 + \sum_{j=1}^{p} x_{ij} x_{ij}'\right)^d \tag{3}$$

Classification results of the combination of non-linear kernel and support vector classifier are called the SVM (Equation (3)).

The SVM classifier, which is particularly designed for binary classification, is a kernel-based supervised learning algorithm that classifies the data into two or more classes and it is not recommended when there are a large number of training examples [8]. A kernel function is a mapping procedure done to the training set to improve its resemblance to a linearly separable data set. The purpose of mapping is to increase the dimensionality of the data set and it is done efficiently using a kernel function. Some of the commonly used kernel functions are linear, RBF, quadratic, Multilayer Perceptron kernel and Polynomial kernel [30]. The

linear kernel function performs well with linearly separable data set and the RBF kernel function performs well with non-linear data set. The linear kernel function takes less time to train the SVM compared with the RBF kernel function. The linear kernel function is also less prone to overfitting compared with the RBF kernel function [31].

The performance of the SVM classifier relies on the choice of the regularization parameter C which is also known as box constraint and the kernel parameter which is also known as the scaling factor. Together they are known as the hyperplane parameter [32]. During the training phase, SVM builds a model, maps the decision boundary for each class and specifies the hyperplane that separates the different classes. Increasing the distance between the classes by increasing the hyperplane margin helps increase the classification accuracy. SVMs can also be used to effectively perform non-linear classification [33].

SVMs have been successfully applied in many diverse fields including text and hypertext categorization [34], image detection, verification, and recognition [35], speech recognition [36], bankruptcy prediction [37], remote sensing image analysis [38], time series forecasting [39], information and image retrieval [40], information security [41], biological *i.e.* bioinformatics and classification of proteins [42] and chemical sciences e.g. data from spectroscopy, *i.e.*, chromatography-mass spectrometry and the neutron magnetic resonance [43].

2.2. *K*-Nearest Neighbor (KNN)

In pattern recognition, the KNN algorithm is an instance based learning method used to classify objects based on their closest training examples in the feature space. An object is classified by a majority vote of its neighbors, that is, the object is assigned to the class that is most common amongst its *k*-nearest neighbors (**Figure 2**), where *k* is a positive integer [44]. In the KNN algorithm, the classification of a new test feature vector is determined by the classes of its *k*-nearest neighbors.

The KNN algorithm is implemented using Euclidean distance metrics to locate the nearest neighbor [45]. The Euclidean distance metrics $d(x,y)$ between two points x and y is calculated using Equation (4).

$$d(x,y)=\sum_{i=1}^{N}\sqrt{x_i^2-y_i^2} \tag{4}$$

where, N is the number of features such that, $x=\{x_1,x_2,x_3,\ldots,x_N\}$ and $y=\{y_1,y_2,y_3,\ldots,y_N\}$.

The KNN classifier is one of the many approaches that attempt to estimate the conditional distribution of Y given X, and then classify a given observation to the class with highest estimated probability. Given a positive integer K and a test observation, x_0, the KNN classifier first identifies the K points in the training data that are closest to x_0, represented by N_0. It then estimates the conditional probability for class j as the fraction of points in N_0 whose response values equal j as indicated in Equation (5):

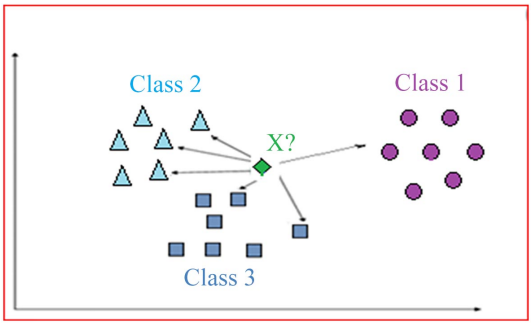

Figure 2. A simple pictorial overview of the K-Nearest Neighbor (KNN) algorithm.

$$\Pr\left(Y=j\middle|X=x_0\right)=\frac{1}{K}\sum_{i\in N_0}I\left(y_i=j\right) \tag{5}$$

where, $I\left(y_i=j\right)$ is an indicator variable that equals 1 if $y_i=j$ and zero if $y_i\neq j$.

KNN is robust to noisy training data and is effective for large numbers of training examples. But for this algorithm, the value of parameter k (number of nearest neighbors) and the type of distance to be used have to be determined. The computation time can be lengthy as one needs to compute the distance of each query instance to all training samples and it gets significantly slower as the number of examples and/or predictors/independent variables increase [24]. Nevertheless, there is no need to build a model, tune several parameters or make additional assumptions. KNN is a simple, versatile, easy to implement supervised MLA that can be used to solve classification, regression and search problems. The algorithm assumes that similar items exist in close proximity. In other words, similar items are near to each other and that 'birds of a feather flock together'. The KNN algorithm hinges on this assumption being true enough for it to be useful [6].

KNN's main disadvantage of becoming significantly slower as the volume of data increases makes it an impractical choice in environments where predictions need to be made rapidly [46]. Moreover, there are faster algorithms that can produce more accurate classification and regression results. However, provided there are sufficient computing resources to speedily handle the data for making predictions, KNN can still be useful in solving problems that have solutions that depend on identifying similar objects [46].

To select the K that is right for a dataset, the KNN algorithm is run several times with different values of K and the K that reduces the number of errors encountered is chosen while maintaining the ability of the algorithm to accurately make predictions when it is applied to data for which it has no prior contact [47]. There are other ways of calculating distance and one way might be preferable depending on the problem that is being solved. However, the straight-line distance, also called the Euclidean distance, is a popular and familiar choice [48].

As the value of K decreases to 1, the predictions become less stable. Inversely,

as the value of K is increased, the predictions become more stable due to majority voting/averaging, and thus, more likely to make more accurate predictions (up to a certain point). Eventually, an increasing number of errors is witnessed. It is at this point that one recognizes that the appropriate value of K has been exceeded. The value of K is usually an odd number to have a tiebreaker in cases where a majority vote among labels is required, for example, picking the mode in a classification problem [49]. The KNN algorithm can be used for classification, regression, and search problems. It is useful in solving problems that have solutions that depend on identifying similar objects.

2.3. Random Forest

Recently there has been a lot of interest in ensemble learning, that is, methods that generate many classifiers and aggregate their results. Two well-known methods are boosting [50] and bagging [51] of classification trees. In boosting, successive trees give extra weight to points incorrectly predicted by earlier predictors. In the end, a weighted vote is taken for prediction. In bagging, successive trees do not depend on earlier trees, each is independently constructed using a bootstrap sample of the data set. In the end, a simple majority vote is taken for prediction [10].

An RF classifier consists of a number of trees, with each tree grown using some form of randomization (**Figure 3**). The leaf nodes of each tree are labeled by estimates of the posterior distribution over the image classes. Each internal node contains a test that best splits the space of data to be classified [17]. An image is classified by sending it down every tree and aggregating the reached leaf distributions. Randomness can be injected at two points during training: in sub-sampling the training data so that each tree is grown using a different subset, and in selecting the node tests [11].

The number of trees necessary for good performance grows with the number of predictors. The best way to determine how many trees are necessary is to compare predictions made by a forest to predictions made by a subset of a forest. When the subsets work as well as the full forest, it indicates there are enough trees. For selecting, m_{try}, Breiman [10] suggests trying the default, half of the default, and twice the default, and then select the best. If one has a very large number of variables but expects only very few to be "important", using a larger m_{try} may give better performance. A lot of trees are necessary to get stable estimates of variable importance and proximity. Since the algorithm falls into the "embarrassingly parallel" category, one can run several random forests on different machines and then aggregate the votes components to get the final result [52].

The RF classifier adds an additional layer of randomness to bagging [10]. In addition to constructing each tree using a different bootstrap sample of the data, RFs change how the classification or regression trees are constructed. In standard trees, each node is split using the best split among all variables while in an RF, each node is split using the best among a subset of predictors randomly

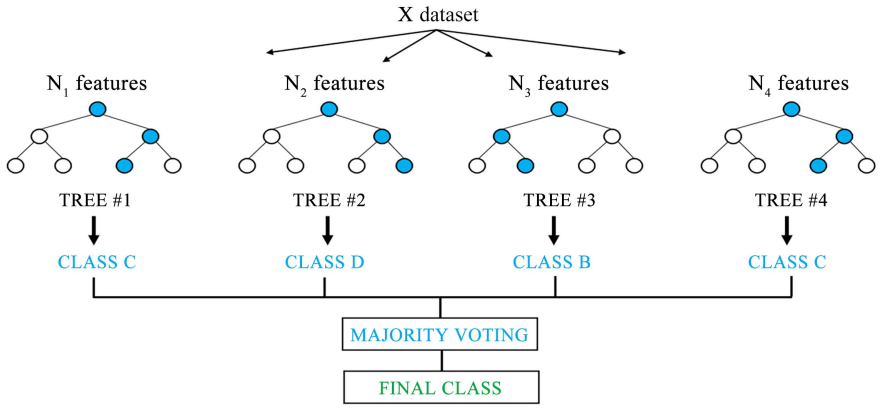

Figure 3. A pictorial overview of the random forest (RF) algorithm.

chosen at that node [53]. This somewhat counterintuitive strategy turns out to perform very well compared with many other classifiers, including discriminant analysis, SVMs and NNs, and is robust against overfitting [4] [10]. In addition, RF is very user-friendly in the sense that it has only two parameters (the number of variables in the random subset at each node and the number of trees in the forest), and is usually not very sensitive to their values [9] [54].

RF is essentially, a set of DTs combined where each tree votes on the class assigned to a given sample, with the most frequent answer winning the vote [55]. This algorithm can handle categorical features very well, can also handle high dimensional spaces as well as a large number of training examples [10]. RF are quite versatile and hence their popularity and application in diverse fields. A decision tree is a set of conditions organized in a hierarchical structure. It is a predictive model in which an instance is classified by following the path of satisfied conditions from the root of the tree until reaching a leaf, which will correspond to a class label. A DT can easily be converted to a set of classification rules [16].

The following types of scientific and engineering data are amenable to RF: DNA data, micro-array data, spectral data: NMR chemical data and molecular structure prediction, quality assessment of manuscripts published in a particular journal, finding clusters of patients based on, for example, tissue marker data, symptoms of a particular disease among others.

2.4. Neural Networks

A NN classifier can be described as a parallel computing system consisting of an extremely large number of simple processors with interconnections [56] [57]. One commonly used type of neural network is a multilayered feed-forward perceptron that consists of several layers of neurons connected with each other (**Figure 4**). The multilayered perceptron can separate data that are nonlinear and generally consists of three or more types of layers [58].

McCulloch and Pitts [59] are generally credited as the designers of the first neural network and earliest mathematical models. Many of their ideas, like many simple units combine to give increased computational power and the idea of a threshold are still used today. The first learning rule on NN was developed on

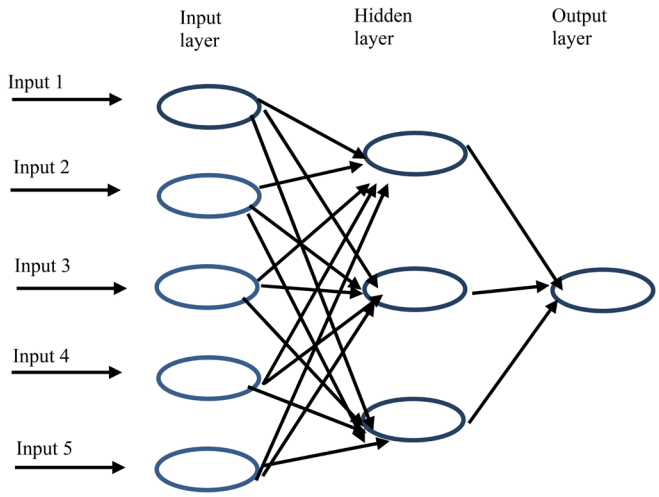

Figure 4. A depiction of a neural network (NN) structure.

the premise that if two neurons were active at the same time the strength between them should be increased [60]. Further improvements and simulations were achieved [61]. During the decades of 1950 and 1960, many researchers worked on the perceptron amidst great excitement, however, by the year 1969, enthusiasm for NN research had waned [62]. Interest for NN research was rekindled in the mid-1980's rekindling [63]. Because of their ability to reproduce and model nonlinear processes, NN have found applications in a wide area of sectors: computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnosis: data mining, cancers, including lung cancer, prostate cancer, colorectal cancers, quantum chemistry among others.

3. Discussion

3.1. Assessing the Performance of a Model

With classification, it is sometimes necessary to use accuracy to assess the performance of a model. Consider analyzing a highly imbalanced data set. For example, trying to determine if a transaction is fraudulent or not, but only 0.5% of the data set contains a fraudulent transaction. Then one could predict that none of the transactions will be fraudulent and have a 99.5% accuracy score which is very misleading. So usually the sensitivity and specificity are used. Using the fraud detection problem, the sensitivity is the proportion of fraudulent transactions identified as fraudulent. The specificity is the proportion of non-fraudulent transactions identified as non-fraudulent.

Therefore, in an ideal situation, what is required are high sensitivity and specificity, although that might change depending on the context. For example, a bank might want to prioritize a higher sensitivity over specificity to make sure it identifies fraudulent transactions. The ROC curve (receiver operating characteristic) is good to display the two types of error metrics described above. The overall performance of a classifier is given by the area under the ROC curve (AUC). Ideally,

it should hug the upper left corner of the graph, and have an area close to 1.

3.2. Attributes of the Classification Algorithms

KNN classifies data based on the distance metric whereas SVM need a proper phase of training. Due to the optimal nature of SVM, it is guaranteed that the separated data would be optimally separated [9]. Generally, KNN is used as multi-class classifiers whereas standard SVM separate binary data belonging to one class or the other. Although, SVMs look more computationally intensive, once training of data is done, that model can be used to predict classes even when applied to new unlabeled data [52]. However, in KNN, the distance metric is calculated each time a set of new unlabeled data is introduced. Hence, in KNN the distance metric always has to be defined [16]. SVMs have two major cases in which classes might be linearly separable or non-linearly separable [46]. When the classes are non-linearly separable, a kernel function such as Gaussian basis function or polynomials is used. Hence, in KNN, only the K parameter have to be set and the distance metric suitable for classification selected whereas in SVMs the R parameter (Regularization term) and also the parameters for kernel if the classes are not linearly separable have to be selected [8]. A main advantage of SVM classification is that it performs well on datasets that have many attributes, even when there are only a few cases that are available for the training process [7]. However, several disadvantages of SVM classification include limitations in speed and size during both training and testing phase of the algorithm and the selection of the kernel function parameters [53].

KNN is easy to implement and understand, but has a major drawback of becoming significantly slow as the size of that data in use increases [24]. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number of examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression) [5]. In the case of classification and regression, choosing the right K for a set of data is done by trying several Ks and picking the one that works best. However, KNN is less computationally intensive and easy to implement than SVM hence it is mostly used in the classification of multi-class data [15]. The algorithm that guarantees reliable detection in unpredictable situations depends upon the data. If the data points are heterogeneously distributed, both KNN and SVM work well [18] [64]. For homogeneous data, one might be able to classify better by putting in a kernel into the SVM. For most practical problems, KNN is a bad choice because it scales badly, if there are a million labelled examples, it would take a long time (linear to the number of examples) to find K nearest neighbors [14].

Different factors affect the capacity of NN to generalize, that is, to predict new data from the learning carried out with training data. The intrinsic factors to network design include the number of neurons and network architecture [6]. The problem of how to define the most suitable network architecture is related to the nature of the hidden layer. There is no rule for determining the number of

hidden layers, but, theoretically, one single hidden layer can represent any Boolean function [65]. In general terms, the higher the number of units of the hidden layer, the greater the NN capacity to represent the training data patterns. However, the fact that the hidden layer has a high number of units also produces a loss in the networks' generalization power [4] [65] [66].

Unlike most methods based on machine learning, RF only needs two parameters to be set for generating a prediction model, that is, the number of regression trees and the number of evidential features (m) which are used in each node to make regression trees grow [19]. It has been demonstrated that with RF, by increasing the number of trees the generalization error always converges; hence, overtraining is not a problem [51]. On the other hand, reducing the number of m brings as a result a reduction in the correlation among trees, which increases the model's accuracy [49].

Adding more data would lengthen NN training times to unacceptable levels so that it would be highly impractical to work with them. Larger input datasets will lengthen classification times for NN more than for SVM and RF [4]. NN has the potential to become a more widely used classification algorithm, but because of their time-consuming parameter tuning procedure, the numerous types of neural network architectures to choose from, and the high number of algorithms used for training NN, some researchers recommend SVM or RF as easier methods which repeatedly achieve results with high accuracies and are often faster [67] [68].

The performance characteristics and attributes of the four types of non-parametric classification algorithms are summarized in **Table 1**.

Table 1. Comparison of the four non-parametric classification algorithms.

Algorithms	Attributes	
	Positive	**Negative**
SVM	1) Kernel functions such as Gaussian basis function or polynomials aids in non-linear separable classes. 2) Works well as a linear classifier. 3) Performs well on datasets that have many attributes. 4) Guarantees optimal separation of data. 5) Works well with heterogeneous distributed points.	1) Computationally intensive when dealing with unlabeled dataset. 2) Has limitation in speed and size during both training and testing phase of the algorithm. 3) Has limitation in speed with regards to the selection of the kernel function parameters.
KNN	1) Easy to implement and understand. 2) Considered high computational complexity because one needs to calculate Euclidean distance of input feature with all the features in the database. However, it is free of training phase but computational in classification phase. 3) Works well with heterogeneous distributed points.	1) It scales badly, if there are a million labeled examples in the dataset. 2) It would take a long time to find K nearest neighbors when there are a million labeled examples in the dataset.
NN	1) With a higher number of units of the hidden layer, the network capacity becomes greater to represent the training data patterns.	1) Higher hidden layer has a high number of units and produces a loss in the networks' generalization power. 2) More data would lengthen NN training times to unacceptable levels so that it would be highly impractical to work with them. 3) Has time-consuming parameter tuning procedure.
RF	1) Generalization error always converges even with increasing number of trees. 2) It is not easy to overfit to one particular feature. However, overfitting to training data remains a problem. 3) Achieve results often faster.	1) Larger input datasets will lengthen classification times.

4. Conclusions

The assessed algorithms have different difficulties in their training. DT based algorithms (RF) involve a lesser difficulty in their training. This applies to both simple regression trees and ensembles of trees (RF). When the data are very scarce RF show a better performance compared to NN and SVM which become more complex. SVMs are based on different kernel types, according to which the combination of parameters to be optimized is different. However, it should be highly emphasized that no broader generalizations can be made about the superiority of any method for all types of problems as the performance of the methods might vary for other datasets.

RF is too sensitive to small changes in the training dataset and is occasionally unstable and tends to overfit in the model. KNN is easy to implement and understand but has a major drawback of becoming significantly slow as the size of data in use grows while the ideal value of K for the KNN classifier is difficult to set. The NN method contains a high level of complexity in computational processing, causing it to become less popular in classification applications. SVM and RF are insensitive to noise or overtraining, which shows their ability in dealing with unbalanced data. Among the nonparametric methods, SVM and RF are becoming increasingly popular in image classification research and applications. Larger input datasets will lengthen classification times for NN and KNN more than for SVM and RF. NN has the potential to become a more widely used classification algorithm, but because of their time-consuming parameter tuning procedure, the numerous types of neural network architectures to choose from and the high number of algorithms used for training NN, most researchers recommend SVM or RF as easier methods which repeatedly achieve results with high accuracies and are often faster.

Declarations

Authors' Contributions

The idea was developed by EYB and JO. Literature was reviewed by all authors. All authors contributed to manuscript writing and approved the final manuscript.

Acknowledgements

We thank the anonymous reviewers whose comments made this manuscript more robust.

Funding

This study attracted no funding.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Hastie, T., Tibshirani, R. and Friedman, J. (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition, Springer-Verlag, New York.

[2] Breiman, L. and Ihaka, R. (1984) Nonlinear Discriminant Analysis via Scaling and ACE. Department of Statistics, University of California, Berkeley. https://digitalassets.lib.berkeley.edu/sdtr/ucb/text/40.pdf

[3] McCloskey, M. and Cohen, N.J. (1989) Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation*, **24**, 109-165. https://doi.org/10.1016/S0079-7421(08)60536-8

[4] Brown, W.M., Gedeon, T.D., Groves, D.I. and Barnes, R.G. (2000) Artificial Neural Networks: A New Method for Mineral Prospectivity Mapping. *Australian Journal of Earth Sciences*, **47**, 757-770. https://doi.org/10.1046/j.1440-0952.2000.00807.x

[5] Rigol-Sanchez, J.P., Chica-Olmo, M. and Abarca-Hernandez, F. (2003) Artificial Neural Networks as a Tool for Mineral Potential Mapping with GIS. *International Journal of Remote Sensing*, **24**, 1151-1156. https://doi.org/10.1080/0143116021000031791

[6] Porwal, A., Carranza, E.J.M. and Hale, M. (2004) A Hybrid Neuro-Fuzzy Model for Mineral Potential Mapping. *Mathematical Geology*, **36**, 803-826. https://doi.org/10.1023/B:MATG.0000041180.34176.65

[7] Boser, B.E., Guyon, I.M. and Vapnik, V.N. (1992) A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, July 1992, 144-152. https://www.svms.org/training/BOGV92.pdf https://doi.org/10.1145/130385.130401

[8] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. https://doi.org/10.1007/BF00994018

[9] Abedi, M., Norouzi, G.H. and Bahroudi, A. (2012) Support Vector Machine for Multi-Classification of Mineral Prospectivity Areas. *Computers & Geosciences*, **46**, 272-283. https://doi.org/10.1016/j.cageo.2011.12.014

[10] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**, 5-32. https://doi.org/10.1023/A:1010933404324

[11] Rodriguez-Galiano, V.F. and Chica-Rivas, M. (2014) Evaluation of Different Machine Learning Methods for Land Cover Mapping of a Mediterranean Area Using Multi-Seasonal Landsat Images and Digital Terrain Models. *International Journal of Digital Earth*, **7**, 492-509. https://doi.org/10.1080/17538947.2012.748848

[12] Gutiérrez, S.L.M., Rivero, M.H., Ramírez, N.C., Hernández, E. and Aranda-Abreu, G.E. (2014) Decision Trees for the Analysis of Genes Involved in Alzheimer's Disease Pathology. *Journal of Theoretical Biology*, **357**, 21-25. https://doi.org/10.1016/j.jtbi.2014.05.002

[13] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984) Classification and Regression Trees. Vol. 37, Wadsworth, New York, 237-251. https://rafalab.github.io/pages/649/section-11.pdf

[14] Al-Anazi, A. and Gates, I.D. (2010) A Support Vector Machine Algorithm to Classify Lithofacies and Model Permeability in Heterogeneous Reservoirs. *Engineering Geology*, **114**, 267-277. https://doi.org/10.1016/j.enggeo.2010.05.005

[15] Ben-Hur, A., Horn, D., Siegelmann, H. and Vapnik, V.N. (2001) Support Vector Clustering. *Journal of Machine Learning Research*, **2**, 125-137.

http://www.jmlr.org/papers/volume2/horn01a/horn01a.pdf

[16] Waske, B. and Braun, M. (2009) Classifier Ensembles for Land Cover Mapping Using Multitemporal SAR Imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, **64**, 450-457. https://doi.org/10.1016/j.isprsjprs.2009.01.003

[17] Chen, W., Xie, X.S., Wang, J.L., Pradhan, B., Hong, H.Y., Bui, D.T., Duan, Z. and Ma, J.Q. (2017) A Comparative Study of Logistic Model tree, Random Forest, and Classification and Regression Tree Models for Spatial Prediction of Landslide Susceptibility. *Catena*, **151**, 147-160.
https://opus.lib.uts.edu.au/handle/10453/119788#
https://doi.org/10.1016/j.catena.2016.11.032

[18] Ghimire, B., Rogan, J., Galiano, V.R., Panday, P. and Neeti, N. (2012) An Evaluation of Bagging, boosting, and Random forests for Land-Cover Classification in Cape Cod, Massachusetts, USA. *GIScience & Remote Sensing*, **49**, 623-643.
https://doi.org/10.2747/1548-1603.49.5.623

[19] Rodriguez-Galiano, V.F., Ghimire, B., Rogan, J., Chica-Olmo, M. and Rigol-Sanchez, J.P. (2012) An Assessment of the Effectiveness of a Random Forest Classifier for Land-Cover Classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, **67**, 93-104. https://doi.org/10.1016/j.isprsjprs.2011.11.002

[20] Pierdicca, R., Malinverni, E.S., Piccinini, F., Paolanti, M., Felicetti, A. and Zingaretti, P. (2018) Deep Convolutional Neural Network for Automatic Detection of Damaged Photovoltaic Cells. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, **42**, 893-900.
https://doi.org/10.5194/isprs-archives-XLII-2-893-2018

[21] Ikeda, K. and Aoishi, T. (2005) An Asymptotic Statistical Analysis of Support Vector Machines with Soft Margins. *Neural Networks*, **18**, 251-259.
https://doi.org/10.1016/j.neunet.2004.11.008

[22] Raczko, E. and Zagajewski, B. (2017) Comparison of Support Vector Machine, Random Forest and Neural Network Classifiers for Tree Species Classification on Airborne Hyperspectral APEX Images. *European Journal of Remote Sensing*, **50**, 144-154. https://doi.org/10.1080/22797254.2017.1299557

[23] Segal, M.R. (2004) Machine Learning Benchmarks and Random Forest Regression. Center for Bioinformatics and Molecular Biostatistics, University of California, San Francisco. https://escholarship.org/uc/item/35x3v9t4

[24] Statnikov, A., Wang, L. and Aliferis, C.F. (2008) A Comprehensive Comparison of Random Forests and Support Vector Machines for Microarray-Based Cancer Classification. *BMC Bioinformatics*, **9**, Article No. 319.
https://doi.org/10.1186/1471-2105-9-319

[25] Bosch, A., Zisserman, A. and Munoz, X. (2007) Image Classification Using Random Forests and Ferns. 2007 *IEEE* 11*th International Conference on Computer Vision*, Rio de Janeiro, 14-21 October 2007, 1-8.
https://doi.org/10.1109/ICCV.2007.4409066

[26] Zuo, R.G. and Carranza, E.J.M. (2011) Support Vector Machine: A Tool for Mapping Mineral Prospectivity. *Computers & Geosciences*, **37**, 1967-1975.
https://www.academia.edu/32294845/
https://doi.org/10.1016/j.cageo.2010.09.014

[27] Sluiter, R. and Pebesma, E.J. (2010) Comparing Techniques for Vegetation Classification Using Multi- and Hyperspectral Images and Ancillary Environmental Data. *International Journal of Remote Sensing*, **31**, 6143-6161.
https://doi.org/10.1080/01431160903401379

[28] Huang, C., Davis, L.S. and Townshend, J.R.G. (2002) An Assessment of Support Vector Machines for Land Cover Classification. *International Journal of Remote Sensing*, **23**, 725-749. https://doi.org/10.1080/01431160110040323

[29] Dietrich, R., Opper, M. and Sompolinsky, H. (1999) Statistical Mechanics of Support Vector Networks. *Physical Review Letters*, **82**, 2975. https://doi.org/10.1103/PhysRevLett.82.2975

[30] Suykens, J.A.K. and Vandewalle, J. (1999) Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, **9**, 293-300. https://doi.org/10.1023/A:1018628609742

[31] Maji, S., Berg, A.C. and Malik, J. (2008) Classification Using Intersection Kernel Support Vector Machines is Efficient. 2008 *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 23-28 June 2008, 1-8. https://doi.org/10.1109/CVPR.2008.4587630

[32] Shen, T., Li, H.S., Qian, Z. and Huang, X.L. (2009) Active Volume Models for 3D Medical Image Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 20-25 June 2009, 707-714. https://doi.org/10.1109/CVPR.2009.5206563

[33] Tsai, C.F., Hsu, Y.F., Lin, C.Y. and Lin, W.Y. (2009) Intrusion Detection by Machine Learning: A Review. *Expert Systems with Applications*, **36**, 11994-12000. https://doi.org/10.1016/j.eswa.2009.05.029

[34] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C. (2002) Text Classification Using String Kernels. *Journal of Machine Learning Research*, **2**, 419-444. http://www.jmlr.org/papers/volume2/lodhi02a/lodhi02a.pdf

[35] Smeraldi, F. and Bigun, J. (2002) Retinal Vision Applied to Facial Features Detection and Face Authentication. *Pattern Recognition Letters*, **23**, 463-475. https://doi.org/10.1016/S0167-8655(01)00178-7

[36] Ganapathiraju, A., Hamaker, J.E. and Picone, J. (2004) Applications of Support Vector Machines to Speech Recognition. *IEEE Transactions on Signal Processing*, **52**, 2348-2355. https://ieeexplore.ieee.org/abstract/document/1315952 https://doi.org/10.1109/TSP.2004.831018

[37] Shin, K.S., Lee, T.S. and Kim, H.J. (2005) An Application of Support Vector Machines in Bankruptcy Prediction Model. *Expert Systems with Applications*, **28**, 127-135. https://doi.org/10.1016/j.eswa.2004.08.009

[38] Melgani, F. and Bruzzone, L. (2004) Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. *IEEE Transactions on Geoscience and Remote Sensing*, **42**, 1778-1790. https://ieeexplore.ieee.org/document/1323134 https://doi.org/10.1109/TGRS.2004.831865

[39] Kim, K.J. (2003) Financial Time Series Forecasting Using Support Vector Machines. *Neurocomputing*, **55**, 307-319. https://doi.org/10.1016/S0925-2312(03)00372-2

[40] Liu, Y.H. and Chen, Y.T. (2007) Face Recognition Using Total Margin-Based Adaptive Fuzzy Support Vector Machines. *IEEE Transactions on Neural Networks*, **18**, 178-192. https://pubmed.ncbi.nlm.nih.gov/17278471/ https://doi.org/10.1109/TNN.2006.883013

[41] Mukkamala, S., Janoski, G. and Sung, A. (2002) Intrusion Detection Using Neural Networks and Support Vector Machines. *Proceedings of the* 2002 *International Joint Conference on Neural Networks*, Honolulu, 12-17 May 2002, 1702-1707.

[42] Zhou, X. and Tuck, D.P. (2007) MSVM-RFE: Extensions of SVM-RFE for Multiclass Gene Selection on DNA Microarray Data. *Bioinformatics*, **23**, 1106-1114. https://pubmed.ncbi.nlm.nih.gov/17494773/

https://doi.org/10.1093/bioinformatics/btm036

[43] Ivanciuc, O. (2007) Applications of Support Vector Machines in Chemistry. *Reviews in Computational Chemistry*, **23**, 291. https://doi.org/10.1002/9780470116449.ch6

[44] Hmeidi, I., Hawashin, B. and El-Qawasmeh, E. (2008) Performance of KNN and SVM Classifiers on Full Word Arabic Articles. *Advanced Engineering Informatics*, **22**, 106-111. https://doi.org/10.1016/j.aei.2007.12.001

[45] Pan, F., Wang, B.Y., Hu, X. and Perrizo, W. (2004) Comprehensive Vertical Sample-Based KNN/LSVM Classification for Gene Expression Analysis. *Journal of Biomedical Informatics*, **37**, 240-248. https://doi.org/10.1016/j.jbi.2004.07.003

[46] Halvani, O., Steinebach, M. and Zimmermann, R. (2013) Authorship Verification via K-Nearest Neighbor Estimation. Notebook PAN at CLEF. *CLEF* 2013 *Working Notes*, Valencia, 23-26 September 2013. http://publica.fraunhofer.de/documents/N-350740.html

[47] Chen, H.L., Huang, C.C., Yu, X.G., Xu, X., Sun, X., Wang, G. and Wang, S.J. (2013) An Efficient Diagnosis System for Detection of Parkinson's Disease Using Fuzzy *K*-Nearest Neighbor Approach. *Expert Systems with Applications*, **40**, 263-271. https://doi.org/10.1016/j.eswa.2012.07.014

[48] Chan, J.C.W. and Paelinckx, D. (2008) Evaluation of Random Forest and Adaboost Tree-Based Ensemble Classification and Spectral Band Selection for Ecotope Mapping Using Airborne Hyperspectral Imagery. *Remote Sensing of Environment*, **112**, 2999-3011. https://doi.org/10.1016/j.rse.2008.02.011

[49] Vincenzi, S., Zucchetta, M., Franzoi, P., Pellizzato, M., Pranovi, F., De Leo, G.A. and Torricelli, P. (2011) Application of a Random Forest Algorithm to Predict Spatial Distribution of the Potential Yield of *Ruditapes philippinarum* in the Venice Lagoon, Italy. *Ecological Modelling*, **222**, 1471-1478. https://doi.org/10.1016/j.ecolmodel.2011.02.007

[50] Shapire, R.E. and Singer, Y. (1998) BoosTexter: A System for Multi-Label Text Categorization. *Machine Learning*, **39**, 135-168. https://www.cis.upenn.edu/~mkearns/finread/boostexter.pdf

[51] Breiman, L. (1996) Bagging Predictors. *Machine Learning*, **24**, 123-140. https://link.springer.com/content/pdf/10.1007/BF00058655.pdf https://doi.org/10.1007/BF00058655

[52] Edwin, R. and Bogdan, Z. (2017) Comparison of Support Vector Machine, Random Forest and Neural Network Classifiers for Tree Species Classification on Airborne Hyperspectral APEX Images. *European Journal of Remote Sensing*, **50**, 144-154. https://doi.org/10.1080/22797254.2017.1299557

[53] Coimbra, R., Rodriguez-Galiano, V., Olóriz, F. and Chica-Olmo, M. (2014) Regression Trees for Modeling Geochemical Data-An Application to Late Jurassic Carbonates (Ammonitico Rosso). *Computers & Geosciences*, **73**, 198-207. https://doi.org/10.1016/j.cageo.2014.09.007

[54] Wang, Z.L., Lai, C.G., Chen, X.H., Yang, B., Zhao, S.W. and Bai, X.Y. (2015) Flood Hazard Risk Assessment Model Based on Random Forest. *Journal of Hydrology*, **527**, 1130-1141. https://doi.org/10.1016/j.jhydrol.2015.06.008

[55] Sun, L. and Schulz, K. (2015) The Improvement of Land Cover Classification by Thermal Remote Sensing. *Remote sensing*, **7**, 8368-8390. https://doi.org/10.3390/rs70708368

[56] Jain, A.K., Duin, R.P.W. and Mao, J.C. (2000) Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 4-37.

https://doi.org/10.1109/34.824819

[57] Hu, Y.H. and Hwang, J.N. (2001) Handbook of Neural Network Signal Processing. CRC Press, Boca Raton.

[58] Beluco, A., Engel, P.M. and Alexandre, B. (2015) Classification of Textures in Satellite Image with Gabor Filters and a Multi-Layer Perceptron with Back Propagation Algorithm Obtaining High Accuracy. *International Journal of Energy & Environment*, **6**, 437-460.
http://www.beluco.net/papers/2015-beluco-engel-beluco-ijee.pdf
https://doi.org/10.5935/2076-2909.20150001

[59] McCulloch, W.S. and Pitts, W. (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*, **5**, 115-133.
https://doi.org/10.1007/BF02478259

[60] Hebb, D.O. (1949) The Organization of Behavior: A Neuropsychological Theory. Wiley, New York.

[61] Werbos, P. (1974) Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. *Ph. D.* Dissertation, Harvard University, Cambridge.

[62] Minsky, M. and Papert, S.A. (1969) Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge.
https://core.ac.uk/download/pdf/82206249.pdf

[63] Hinton, G., LeCun, Y. and Bengio, Y. (2015) Deep Learning. *Nature*, **521**, 436-444.
https://doi.org/10.1038/nature14539

[64] Palaniappan, R., Sundaraj, K. and Sundaraj, S. (2014) A Comparative Study of the SVM and K-NN Machine Learning Algorithms for the Diagnosis of Respiratory Pathologies Using Pulmonary Acoustic Signals. *BMC Bioinformatics*, **15**, Article No. 223. https://doi.org/10.1186/1471-2105-15-223

[65] Atkinson, P.M. and Tatnall, A.R.L. (1997) Introduction to Neural Networks in Remote Sensing. *International Journal of Remote Sensing*, **18**, 699-709.
https://doi.org/10.1080/014311697218700

[66] Foody, G.M. and Arora, M.K. (1997) An Evaluation of Some Factors Affecting the Accuracy of Classification by an Artificial Neural Network. *International Journal of Remote Sensing*, **18**, 799-810. https://doi.org/10.1080/014311697218764

[67] Petropoulos, G.P., Kalaitzidis, C. and Vadrevu, K.P. (2012) Support Vector Machines and Object-Based Classification for Obtaining Land-Use/Cover Cartography from Hyperion Hyperspectral Imagery. *Computers & Geosciences*, **41**, 99-107.
https://doi.org/10.1016/j.cageo.2011.08.019

[68] Shrivastava, R., Mahalingam, H. and Dutta, N.N. (2017) Application and Evaluation of Random Forest Classifier Technique for Fault Detection in Bioreactor Operation. *Chemical Engineering Communications*, **204**, 591-598.
https://doi.org/10.1080/00986445.2017.1292259