Machine Learning with Applications 4 (2021) 100031

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Review of classification algorithms with changing inter-class distances[☆]

1. Introduction

The world is currently expressing overwhelming data explosion; in volume and complexity. Consequently, data analytics is faced with the challenge of adopting or developing suitable algorithms to extract useful patterns or information from these huge datasets. Data mining has become an efficient and useful approach capable of mitigating the challenge of analysing 'big data' (Asghar & Iqbal, 2009). The main aim of data mining is to extract, discover or "mine" useful patterns which are hidden in the datasets (Rohanizadeh & BAMENI, 2009; Sahu, Shrma, & Gondhalakar, 2011). One of the data mining tools is machine learning (ML); which generally could be supervised, unsupervised or reinforcement learnings (Bishop, 2006; Bradshaw, Hoffman, Woods, & Johnson, 2013; Samuel, 1988; Scharre & Horowitz, 2015). Literature shows several ML algorithms have been developed to solve different data related problems. For instance, many feature selection (FS) algorithms, which could be filter, wrapper or embedded, have been developed to undertake the selection of relevance features for improved data classification (Dasgupta, Drineas, Harb, Josifovski, & Mahoney, 2007; Jović, Brkić, & Bogunović, 2015; Vafaie & Imam, 1994). Interestingly, some of the FS algorithms are capable of correctly selecting the important features only in cases where the importance of the features is uniform or common for all the classes.

However, and regrettably, these algorithms often fail in correctly identifying and selecting the important features in datasets where the input feature relevance is different or not uniform i.e. where a feature could be relevant for one class and irrelevant or noisy for another class. The failure of the FS algorithms to identify this type of feature relevance is due to the inability of the FS algorithms to carry out feature relevance analysis at class level. For any FS algorithm to achieve this, i.e., feature relevance analysis at class level, it must be able to control what features influence the separation of each classes with some sort of weighting functions during training as suggested in Ahmad and Starkey (2018).

In a similar vein, it is particularly of note that, the deep learning which is known for best classification of data including those whose classes are made of multiple relationships, has the challenge of lack of explainablity, as it is considered a black box (Biryulev, Yakymiv, & Selemonavichus, 2010; Darbari, 2000; Hofmann, Schmitz, & Sick, 2003).

Despite the tremendous and explosive research in data science, data analytics and machine learning, it is however notable that researchers have paid less attention to the problem of datasets with varying inter class distances and in some cases combined with other data related problems. This problem is very typical of real-world datasets; considering their complexities therein.

In this regard therefore, this paper seeks to examine some various data related problems and the corresponding performances of eight selected classification algorithms; self-organizing maps (SOM) (Frezza-Buet, 2008; Fritzke, 1995; Kohonen, 1990; Kohonen, Kaski, & Lappalainen, 1997; Prudent & Ennaji, 2005), decision tree (DT) (Lee & Siau, 2001; Ngai, Xiu, & Chau, 2009; Witten, Frank, & Hall, 2005), linear discriminant analysis (LD) (Nikolaou, 0000; Park, Choo, Drake, & Kang, 2008), naïve bayes (NB) (El Kourdi, Bensaid, & Rachidi, 2004; Hristea, 2012; Xu, 2018), support vector machines (SVM) (Cortes & Vapnik, 1995; Liu & Huang, 2002; Nguyen & De la Torre, 2010; Phyu, 2009; Smola & Schölkopf, 2004; Steinwart & Christmann, 2008), k-nearest neighbours (KNN) (Leung, 2007; Mulak & Talhar, 2015; Zhang & Zhou, 2005) and deep learning (DL) (Bengio, Goodfellow, & Courville, 2017; Chen, Lin, Zhao, Wang, & Gu, 2014; Kim & Moon, 2015; Wang, Chen, Xu, & Jin, 2015). Specifically, the data problems studied in this research include: datasets with varying inter class distances (where classes are separated by different amounts); datasets with classes having different input relevance (where a feature could be relevant for one class and noisy for another class); datasets with classes defined by multiple relationships; i.e. the class is effectively made up of sub-classes each defined by a different pattern in the data feature set; datasets with increasing number of noisy features; and datasets with varying amplitudes of noisy features. Furthermore, since real world datasets can be made of a combination of some of the problems stated above, in this research we also synthesized datasets with a combination of some of these problems in order to assess and investigate the performances of the selected algorithms.

2. Background information

Machine learning (ML) is an important subset of Artificial Intelligence which allows computers to learn from and make predictions on data. Machine learning algorithms are able to extract features and common patterns from a set of data (instances or training data), and to apply them to new datasets (test data). Machine learning tasks usually are categorized into supervised learning and unsupervised learning (Bishop, 2006; Bradshaw et al., 2013; Samuel, 1988; Scharre & Horowitz, 2015). In supervised learning, the algorithm learns a function that maps input to output given some example of input–output pairs (Bishop, 2006; Bradshaw et al., 2013; Samuel, 1988; Scharre & Horowitz, 2015). A supervised learning algorithm works on labelled data. In supervised learning, a function is deduced based on a set of training data with labels. The supervised learning algorithm, after analysing the input vectors, infers a function known as classifier or regression function, depending if the output is discrete or continuous. The inferred function must be able to predict the class for any input vector (Ezenkwu & Starkey, 2019).

[☆] This document is the preliminary results of Ph.D. research study carried out at the University of Aberdeen.

https://doi.org/10.1016/j.mlwa.2021.100031
23 November 2020
Available online 22 March 2021

Unlike supervised learning, an unsupervised learning algorithm works on unlabelled data. This implies that in unsupervised learning, there are no output responses (labels) used during the training process, although they can be assigned following completion of training. In this type of learning, nothing is known a priori. It is a way to find hierarchy and order in a set of data without structure. In this method, data is grouped based on the features they share (Bishop, 2006; Bradshaw et al., 2013; Samuel, 1988; Scharre & Horowitz, 2015).

2.1. Decision tree

Decision Tree (DT) is a supervised learning (classification) technique that creates a model which anticipates the value of a target variable depending on input values. Breiman et al. proposed the classification and regression trees (CART) algorithm in their work (Breiman, Friedman, Stone, & Olshen, 1984). This has sprouted a number of tree-based methods in machine learning. The success of the CART framework is inspired by its flexibility and interpretability. This provides a nonparametric tool with nonlinear decision boundaries for both classification and regression problems (Breiman et al., 1984). The DT is a graphical representation of the outcome. In the DT, the roots (nodes) represent the tests and attributes, the branches show the results of the tests and the leaves represent the class distributions. This technique is aimed at uncovering the records or relationships which exist between the datasets and also infer the rules that define or govern these relationships (Lee & Siau, 2001; Ngai et al., 2009; Witten et al., 2005). In the DT, each path from the root (node) of a decision tree to one of its leaves can be converted into a rule by simply combining the tests along the path in order to form the antecedent part, with the leaf's class prediction taken as the class value. Examples of decision tree are the C4.5 and C5.0. The C4.5 algorithm constructs a very large tree based on all input attribute values and finalizes the decision rules by pruning. C5.0 is an algorithm developed from C4.5. The idea of construction of a decision tree in C5.0 is similar to C4.5. Keeping all the functions of C4.5, C5.0 introduces more new techniques such as boosting. Practical implementation of DT algorithm includes the Recursive Partitioning and Regression Trees (Rpart). The Rpart algorithm works by splitting the dataset recursively. This involves further splitting the subsets that arise from a split until a predetermined termination criterion is reached. At each step, the split is made based on the independent variable that results in the largest possible reduction in heterogeneity of the dependent (predicted) variable (Ngai et al., 2009). DTs have the strengths of being self-explanatory, easy to understand and can be converted to a set of rules easily. Also, DTs are capable of handling datasets with errors and missing values. However, the weaknesses of the DT include its ability to work only on datasets with discrete targets and are useful only for non-complex tasks. In addition, DTs have greedy characteristic of being over-sensitive to the training dataset, irrelevant attributes and to noise thereby making them prone to errors when the classes are too many (Ngai et al., 2009).

2.2. Artificial neural networks

Artificial Neural Networks (ANN) also known as Deep Learning (depending on the number of hidden layers) is another supervised learning algorithm. ANN was inspired by research on the human brain system. ANN is a powerful technique of data mining. It is inspired by biological systems capable of detecting patterns and as a result can make predictions (Bengio et al., 2017; Chen et al., 2014; Kim & Moon, 2015; Wang et al., 2015). The major breakthroughs in neural networks in recent years are in their application to real world problems. This includes real life problems of fraud detection and customer response in business. By their special ability to learn, DL is capable of extracting important relationships or patterns from analysis of incomplete, complex and perhaps imprecise data which cannot be detected by other computational methods (Darbari, 2000). An architecture of DL consists of input layer, hidden layers and output layers. Applications show that DL has the strength of higher predictive accuracy obtained than other methods or human experts for some data problems (Jain & Srivastava, 2013). Also, DL has the ability of distributed information storage, parallel processing, reasoning, and self-organization (Jain & Srivastava, 2013). It also has the capability of rapid fitting of nonlinear data, so it can solve many problems which are difficult for other methods (Biryulev et al., 2010). Despite these tremendous strengths, DL lacks transparency. DL algorithm is regarded as black box. This means that its predictions cannot be explained. This is as a result of the complexity of its architecture. As simple as a single hidden layered ANN may appear, it is not possible to explain why a particularly data point is considered a member of a class and another a member of another class (Darbari, 2000). Another limitation of DL is that there are no symbolic rules on how the classification is done. Therefore, it cannot be explicitly suitable for verification and interpretation by human experts (Biryulev et al., 2010; Darbari, 2000).

2.3. Support vector machines

(Boser, Guyon, & Vapnik, 1992) proposed a training algorithm for optimal margin classifiers now known as Support Vector Machines (SVMs). Their attempt was to maximize the margin between the class boundary and training patterns in order to optimize the cost functions such as the mean squared error (MSE). SVM has become very popular since then and is often used for both regression and classification. The SVM aims to find optimal hyperplanes that segregate multiple groups. These hyperplanes could be linear or nonlinear. An SVM classifier tries to identify a hyperplane with the ability of maximizing the distance between the closest points to the hyperplane itself (Cortes & Vapnik, 1995; Liu & Huang, 2002; Nguyen & De la Torre, 2010; Phyu, 2009; Smola & Schölkopf, 2004; Steinwart & Christmann, 2008). SVM is often described as a discriminative classifier defined by data separator line called the hyperplane. When presented with labelled data, the SVM tries to infer an optimal line or lines (hyperplane) which separates new sets of data into the classes they belong. Typically, in a 2D space, the hyperplane separates a plane in two sections with each class on either side. The data points from the different classes are usually separated by a distance between their decision surfaces referred to as the margin. The decision surface is usually a line that links the data points that lie closest to the boundary of the class. These data points are called support vectors. The support vectors are very important in the construction of the SVM classifier. SVM was developed as a linear algorithm initially (Cortes & Vapnik, 1995). Several modifications to handle nonlinearly separable classes have been proposed, including, the kernel trick, which has been used in solving several classification problems as presented in Fu et al. (2004), Hsieh, Chang, Lin, Keerthi, and Sundararajan (2008), Jose, Goyal, Aggrwal, and Varma (2013), Zhang, Lan, Wang, and Moerchen (2012) and Zhu, Liu, Lu, and Li (2016).

The strengths of SVM include its ability to provide a good out-of-sample generalization. This is particularly useful if the training parameters are correctly selected. This causes the outliers in the data to be redundant and consequently have reduced effect on the hyperplane. With this, SVM exhibits robustness for some level of training sample (Meyer, 2004). Another strength of the SVM is the introduction of kernel trick. The introduction of kernel gives SVMs flexibility in choosing thresholds that separate group of datasets which may not have to be linear nor have a similar functional form for all data (Meyer, 2004). The major setback of SVM is lack of Transparency. SVMs are not capable of describing the contribution of individual data features to the identification of the Hyper-plane during training as the result of high dimension. Furthermore, the Choice of Kernel is another challenge of SVM. In (Meyer, 2004), it was demonstrated that the performance of SVMs is a function of the parameters selected at the initial stage of the training. For SVM algorithm to be effective in achieving the best classification results, some major parameters must be correctly set. Some of the parameters may result in a good classification accuracy for a problem but give poor classification accuracy for another. Therefore, the data analyst would have to experiment with different parameter values before achieving a satisfactory and acceptable result. Also, SVM has the disadvantage of being sensitive to irrelevant inputs. Investigation in Aggarwal (2014) proved that SVM algorithm is very sensitive to irrelevant features at the beginning of training. The presence of irrelevant inputs in the datasets can be misleading to the SVM, thereby, resulting in wrong or poor classification, as demonstrated in Nguyen and De la Torre (2010).

2.4. Naïve Bayes

Naïve Bayes (NB), also a supervised learning algorithm was developed based on Bayes theorem as described in Hristea (2012) and Xu (2018). It is a probability-based classification algorithm that classifies data into their classes based on the 'Maximum a Posteriori' decision rule in a Bayesian setting. A major assumption of the NB is that the effect of the value of a variable on a given class is independent of the values of other variables. This is known as conditional independence. NB classifiers are used for text classification, as well as spam detection. NB classifiers work by calculating the probabilities for every factor and selecting the outcome which has the highest probability (El Kourdi et al., 2004). Mathematically, Naïve Bayes is represented as shown in Eq. (1).

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{1}$$

Where; $P(X|Y)$ is the probability of X given Y (i.e. the probability of X given that Y occurs), $P(X)$ is the probability of X, $P(Y|X)$ is the probability of Y given X (i.e. the probability of Y given that X occurs) and $P(Y)$ is the probability of Y.

The NB algorithms have been applied in automated data analysis for automatic text categorization. These are as presented in Hristea (2012). It has been noted from these literatures that these applications have restrictions on specific problems. Also, there are needs for pre-processing methods which are often manually carried out on the document before analysis can be done. The Naïve Bayes has very fast training time convergence time as demonstrated in El Kourdi et al. (2004). The Naïve Bayes method assumes that features are independent (Xu, 2018). However, the assumption of independence of features is a major weakness of the NB because the assumption is not always true for real-world datasets where patterns are usually defined by dependent features of the datasets.

2.5. Linear discriminant analysis

Discriminant analysis was first applied by R. Fisher, an English scientist who developed how species of birds could be classified (Fisher, 1936). He considered group separation involving only two classes. The idea was to find a linear combination of the predictors that shows the largest difference in the group means in relation to the variance within group. Linear Discriminant Analysis (LDA) is commonly used as a dimensionality reduction and classification method and projects data onto a lower dimensional space. This is often done in such a way as to maximize the ratio of inter-class distance to the intra-class distance. This results in achieving maximum discrimination (Ye, Janardan, & Li, 2005). LDAs are known to be fast in training with quick convergence. As reported in Park et al. (2008), LDA assumes unimodal Gaussian likelihoods. This is a major weakness in the LDAs. Assuming significantly non-Gaussian distributions, the LDAs projections will fail to preserve the complex structure in the data that is required for classification (Park et al., 2008). LDA also fails on complex datasets where discriminatory information is in the variance and not in the mean of the data (Nikolaou, 0000).

2.6. K-Nearest Neighbours

K-Nearest Neighbours (KNN) algorithm is another supervised learning method. KNN makes assumption that similar things exist in close proximity which implies that similar things are close to each other (Leung, 2007; Mulak & Talhar, 2015). KNN finds the distances between a selected data sample with all the samples in the dataset and selecting the specified number of samples (K) closest to the selected data sample. The most frequent label, if it is for classification is then considered the winning class for the data sampled selected (Zhang & Zhou, 2005). KNN is easy to use and has quick calculation time. Also, KNN does not make assumptions about the data. However, KNN's accuracy depends on the quality of the data. Another challenge associated with KNN is the difficulty in finding an optimal value for k. KNN has poor classification accuracy at classifying data points in a boundary where they can be classified one way or another (Zhang & Zhou, 2005).

2.7. Self-organizing maps

Self-organizing Maps (SOM) are used for projection of data in low dimensional spaces, usually 2-dimension (2-D). Model 1005 was proposed by Kohonen in Kohonen (1990). This model consists of a set of C discrete cells known as the "map". This map is made of a discrete topology which is defined by two dimensional graphs. A SOM architecture consists of two layers; the input and computational layers (Kamruzzaman & Sarkar, 2011). The input layer usually is made of the source nodes which represents the input features (Deboeck & Kohonen, 2001; Kohonen, 1990; Kohonen et al., 1997; Lebbah, Rogovschi, & Bennani, 2007). According to Kohonen (1990), the determination of the set of weights (W) parameters is by minimizing the cost function iteratively as shown in Eq. (2).

$$R(C,W) = \sum_{i=1}^{N} \sum_{j=1}^{|W|} K_{j,c(x)} \| X_i - W_j \|^2 \tag{2}$$

Where $K_{j,c(x)(t)}$ is further expressed as;

$$K_{j,c(x)(t)} = \exp(\frac{-\delta_{j,c}^2}{2\sigma^2(t)} t = 0, 1, 2, \ldots ..) \tag{3}$$

for

$$t = 0, 1, 2, \ldots .. \tag{4}$$

where $K_{j,c(x)(t)}$ is known as the neighbourhood function between each of the unit (j) on the map and the winning unit $C(x_i)$ at the tth training step, $\delta_{j,c(x)(t)}$ is known as the distance usually Euclidean, between unit (j) and the winning unit $C(x_i)$ on the map and σ(t) represents the effective width of the topological neighbourhood at the training step tth. The strength of SOM is in its learning transparency. SOM's training is quite explainable and transparent. This can be done by assessing and studying the input mapping change at every epoch with the best matching unit (BMU) node (Kohonen, 1990; Lebbah et al., 2007). One of the weaknesses of SOM is that all input features are given the same level of relevance. During training, SOM treats all features with equality and does not account for the contribution of the features after training. Another limitation is in its manual initialization of number of nodes. SOM requires a pre-defined size of SOM dimension as input parameter. This poses a problem when the underlying data complexity exceeds the SOM dimension or where there is no prior knowledge of the change in the dynamic distribution of the inputs (Westerlund, 2005).

2.8. Growing neural gas

To overcome the limitation of pre-defined size of SOM dimension, the Growing neural gas (GNG) was proposed in (Fritzke, 1995). The idea behind the GNG is to grow the nodes by successively adding new nodes to an initially small network through evaluation of local statistical measures that were gathered during previous adaptation steps. The

GNG grows during the training to find the appropriate size (number of nodes) for a given data (Fritzke, 1995). According to Ezenkwu and Starkey (2019), the GNG has an advantage over the standard SOM in the sense that, in GNG the correct number of nodes is not expected to be decided a priori. For this reason, the GNG is suitable in cases of unknown distribution of sample observations. The key challenge of GNG is in the mechanisms for deciding when to add a new node. The GNG adds nodes at constant number of intervals. In order to overcome this, a ceiling is usually provided to indicate when to stop adding (growing) nodes.

2.9. Random forest

The Random forest (rf) classifier, as the name implies, consists of a large number of individual decision trees which operate as an ensemble. In this algorithm, each individual tree in the random forest gives a class prediction and the class with the most votes among the trees, becomes the model's predicted class. The fundamental concept and motivation behind rf is the idea of the wisdom of crowds. This stems from the idea that a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual tree models. In other words, this implies that a group of "weak learners" (trees), if combined together can build a "strong learner". Like DT classifier, rf classifier does not need feature scaling. Unlike DT classifier, rf classifier is more robust to the selection of training samples and noise in training dataset. However, the rf classifier is harder to interpret (Mishra & Suhas, 2016); (Chu et al., 2014; Nguyen, Wang, & Nguyen, 2013).

3. Review of related works

Rado, Ali, Sani, Idris, and Neagu (2019) carried out an evaluation of the performance of five classification algorithms; C5.0, Rpart, k-nearest neighbour (KNN), support vector machines (SVM), and random forest (RF), with correlation-based feature selection, variables importance selection, and recursive feature elimination selection techniques on some relevant numerical and mixed healthcare datasets. Their experiment showed that some classification algorithms could not yield promising classification results due to a lack of feature selection capability of the classification algorithms. However, when applied with feature selection algorithm, Rpart, the classification algorithms were observed to have improvements in classification.

In Zaffar, Hashmani, and Savita (2017), Zaffar et al. carried out performance analysis of feature selection algorithms on student dataset with the aim of aiding researchers find the best combinations of feature selection (FS) algorithms and classifiers.

An illustrative instance on how some existing feature selection techniques can be integrated into a classification algorithm in order to take advantage of individual algorithms is presented in Liu and Yu (2005). With a well-defined categorizing framework, the researchers built an integrated system for intelligent feature selection with a unifying platform as an intermediate step.

The problem of class imbalance in datasets which is equally one of the key challenges in ML was examined by Sui, Zhang, Huan, and Hong (2019). The aim of their study was to study different sampling methods and their abilities to enhance improvement in classification performance on datasets with imbalanced classes. In particular, they investigated ten sampling methods. From their study, it was shown that imbalanced datasets resulted in sub-optimal performance of the classification algorithms under investigation. In order to address this problem, several data sampling techniques have been proposed. However, they concluded that, there seems to be no universally stand-alone solution to this problem, therefore the need to explore many data sampling techniques in order to decide which is more efficient in balancing class distribution.

Table 1
Average distances between classes for Setup 1: Equally separated classes.

C_4	24.3	16.3	8.3	0
C_3	16.2	8.2	0	8.3
C_2	8.1	0	8.3	16.3
C_1	0	8.1	16.2	24.3
	C_1	C_2	C_3	C_4

Table 2
Average distances between classes for Setup 2: Unequally separated classes.

C_4	18.1	16.0	8.1	0
C_3	10.0	8.0	0	8.1
C_2	2.1	0	8.0	16.0
C_1	0	2.1	10.0	18.1
	C_1	C_2	C_3	C_4

Table 3
Average distances between classes for Setup 3: Equally separated classes.

C_6	80.2	64.4	47.9	32.4	16.6	0
C_5	63.6	47.9	31.4	15.9	0	16.6
C_4	47.8	32.1	15.6	0	15.9	32.4
C_3	32.3	16.5	0	15.6	31.4	47.9
C_2	15.8	0	16.5	32.1	47.9	64.4
C_1	0	15.8	32.3	47.8	63.6	80.2
	C_1	C_2	C_3	C_4	C_5	C_6

Similar to Sui et al. (2019), an investigation aimed at improving the classification of KNN algorithm for imbalanced data was carried out in Shi (2020). Unlike (Sui et al., 2019) which considered many classification algorithms, (Shi, 2020) only considered KNN as the classification algorithm. With 14 real-world datasets obtained from different domains; web usage records and medical data, the researcher, performed statistical tests in order to discover the significance of some data pre-processing methods required to improve KNN classification on imbalanced datasets. Specifically, among the sampling techniques considered were; random sampling, synthetic minority class oversampling, Wilsons editing, cluster-based sampling, and ensemble data sampling.

Xue and Hauskrecht in Xue and Hauskrecht (2019) proposed an ML framework with the ability to learn multi-class classification models. In this work, the researchers developed techniques for learning multi-class classification models from examples associated with an ordered class set information. In addition, they developed an active learning technique that considers feedback and further evaluated the importance of the proposed framework on multiple datasets patterns. It was illustrated in this paper that the class-order feedback and active learning has the ability to reduce the annotation cost separately or combined.

4. Experimental design

4.1. Datasets

In order to achieve the purpose of this study, synthetic datasets with problems or characteristics outlined earlier were created. The use of the synthetic datasets is necessary and important because they allow for a full assessment and evaluation of the selected algorithms on how well they perform, what data problem can be solved by them, and expose where they may encounter difficulties if any. It also allows us to define exactly the type of data problem to be tested by the various algorithms.

Furthermore, Synthetic datasets can exhibit the attributes of real-world datasets in terms of different data shapes, forms and presence of noise, un-equal input variance and overlapping class definitions etc. However, while the synthetic datasets can mimic many attributes of real-world datasets, they do not copy the original content of the real-world datasets exactly.

Table 4
Average distances between classes for Setup 4: Unequally separated classes.

C_6	72.4	94.4	51.8	34.2	18.7	0
C_5	55.8	52.8	35.2	17.6	0	16.7
C_4	38.2	35.2	17.6	0	17.6	34.2
C_3	20.6	17.6	0	17.6	35.2	51.8
C_2	3.2	0	17.6	35.2	52.8	69.4
C_1	0	3.2	20.6	38.2	55.8	72.4
	C_1	C_2	C_3	C_4	C_5	C_6

4.2. Brief description of datasets

Ten (10) synthetic datasets were generated and used for the experiments. These datasets are generated from a reference relationship for each class and each sample has randomly added noise using a uniform distribution of up to 20% of the reference value. This is in order to make it more difficult for the ML than the normal distribution of noise to be added to the relationships. Setup 1 consists of four classes (equally separated, shown in Table 1) with 50 samples per class and 4 input features. Setup 2 consists of four classes (unequally separated, as shown in Table 2) with 50 samples per class and 4 input features. Setup 3 consists of 6 classes (equally separated, as shown in Table 3) with equal sample distribution of 176 per class and 16 features. Setup 4 consists of 6 classes (unequally separated, as shown in Table 4) with equal sample distribution of 176 per class and 16 features. Setup 5 contains 35 features and 6 classes with unequal distribution of samples and unequal input relevance. This means that classes in this dataset are defined by features independently. Setup 6 contains 23 features and 6 classes with unequal distribution of samples. This dataset has its classes consisting of multiple relationships. Setup 7 consists of 35 features and 6 classes with unequal distribution of samples. It has a combined property of classes defined by different input relevance and multiple relationships. Setup 8 contains 53 features and 6 classes with equal distribution of 200 samples in each class. Setup 9 contains 116 features and 6 classes with equal distribution of 200 samples in each class. Finally, Setup 10 contains 1,016 features and 6 classes with equal distribution of 200 samples in each class. Details of the datasets are presented in Table 5.

5. Results and discussion

The results of the experiments are shown in Table 6. The experiments were repeated 12 times for the datasets on each algorithm and the mean and standard error computed. Discussions of the results for each of the data problems studied in this research are given in the following sections. For clarity, Setup 2 has 4 classes which explains why there are dashes for the classification accuracies for classes 5 and 6 in Table 6.

5.1. Varying inter class distances

Datasets with varying inter class distances i.e. Setup 1, Setup 2, Setup 3 and Setup 4 were investigated and tested on the 8 selected learning algorithms. These algorithms include, SOM, GNG, DT, LD, NB, SVM, KNN and DL. The performances of the 8 algorithms on the datasets with varying inter class distances are shown in Table 6 and Fig. 1. In the experiment for Setup 1, all the selected algorithms gave 100% classification accuracy for all classes of the Setup 1. This was as

Table 5
Datasets and Properties.

Dataset name	Properties and description.
Setup 1	The dataset is made of 4 classes with equal distribution of 50 samples in each class. All classes are defined by all 4 related features and equal class separations. Total samples: 200 Features: 4 Classes: 4.
Setup 2	The dataset is made of 4 classes with equal distribution of 50 samples in each class. All classes are defined by all 4 related features and unequal class separations (varying inter class distances). Total samples: 200 Features: 4 Classes: 4
Setup 3	The dataset contains 16 features and 6 classes with equal distribution of 176 samples in each class. All classes are defined by all 16 related features and equal class separations. Total sample: 1056 Features: 16 Classes: 6.
Setup 4	The dataset contains 16 features and 6 classes with equal distribution of 176 samples in each class and unequal class separations (varying inter class distances). Total sample: 1056 Features: 16 Classes: 6.
Setup 5	The dataset contains 35 features and 6 classes with unequal distribution of samples and unequal input relevance. The datasets were created to evaluate the ability of the selected algorithms to identify irrelevant inputs from datasets with un-equal input variance, defining classes of un-equal distribution. Classes defined by features independently as follows; Class 1 contains 500 samples with relevance features of inputs 1,2, 3,18,21and 30. Class 2 contains 350 samples with relevance features of inputs 1, 2, 3, 10, 15 and 18. Class 3 contains 200 samples with relevance features of inputs 2, 4, 5 11,and 15. Class 4 has 100 samples with relevance features of inputs 2,6,7,23,24,26 and 35. Class 5 contains 80 samples with relevance features of inputs 4,5,6,7,9,22,30 and 35. Class 6 is made of 120 samples with relevance features of inputs of 1,2,3,4,8,21,23,24,25,29,31 and 12 irrelevant features 12,13,14,16,17,19,20,27,28,32,33,34 Total sample: 1350 Features: 35 Classes: 6.
Setup 6	The dataset contains 23 features and 6 classes with unequal distribution of samples and classes with multiple relationships. This is a clean dataset with no noise, irrelevant inputs, no outliers. Total sample: 1350 Features: 23 Classes: 6.
Setup 7	The dataset contains 35 features and 6 classes with unequal distribution of samples with combined properties of classes defined by different input relevance and classes with multiple relationships. Total sample: 1350 Features: 35 Classes: 6.
Setup 8	The dataset contains 53 features and 6 classes with equal distribution of samples in each class. This dataset was obtained from Setup 4 by adding 10 noisy features to it. Total sample: 1200 Features: 53 Classes: 6.
Setup 9	The dataset contains 116 features and 6 classes with equal distribution of samples in each class. This dataset was obtained from Setup 4 by adding 100 noisy features to it. Total sample: 1200 Features: 116 Classes: 6.
Setup 10	The dataset contains 1042 features and 6 classes with equal distribution of samples in each class. This dataset was obtained from Setup 4 by adding 1000 noisy features to it. Total sample: 1200 Features: 1042 Classes: 6.

Table 6
Experimental Results for all datasets apart from Setup 1 and Setup 3 where all methods achieved 100% accuracy.

DATA PROBLEMS WITH DATASETS USED		Accuracy % (Mean ± Standard Error)						
	ALGs	CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	Overall Acc
Varying Inter Class Distance Setup 2 made up of unequally Separated classes of 4 Classes	SOM	65.2±2.60	64.3±2.7	99.5±0.30	99.5±0.30	–	–	82.1±1.4
	GNG	99.2±0.34	98.2±0.31	100.0±0.00	100.0±0.00	–	–	99.4±0.17
	DT	57.4±1.20	66.7±0.60	95.0±0.45	95.0±0.88	–	–	78.5±0.55
	LD	68.5±2.11	66.2±2.30	100.0±0.00	100.0±0.00	–	–	83.7±0.00
	NB	66.2±2.22	68.3±2.21	100.0±0.00	100.0±0.00	–	–	82.6±0.56
	SVM	66.1±2.01	70.2±2.01	100.0±0.00	100.0±0.00	–	–	84.1±0.34
	KNN	55.2±2.28	43.3±2.27	100.0±0.00	100.0±0.00	–	–	74.6±0.88
	DL	67.2±2.41	67.3±2.36	100.0±0.00	100.0±0.00	–	–	83.6±0.88
Varying inter class distance: unequally separated setup 4 made up of classes of 4 classes	SOM	51.7±0.98	52.2±0.97	100.0±0.00	100.0±0.00	82.4±1.96	81.8±1.96	77.8±0.63
	GNG	98.2±0.23	98.2±0.23	100.0±0.00	100.0±0.00	100.0±0.00	100.0±0.00	99.4±0.01
	DT	72.7±1.2	76.4±0.99	100.0±0.00	100.0±0.00	96.0±1.2	98.8±2.16	90.7±1.1
	LD	62.7±0.88	83.3±0.91	100.0±0.00	100.0±0.00	100.0±0.00	88.8±1.36	89.1±0.59
	NB	89.7±0.96	70.3±0.94	100.0±0.00	100.0±0.00	100.0±0.00	81.8±1.96	90.3±0.67
	SVM	84.4±0.87	76.1±0.94	100.0±0.00	100.0±0.00	100.0±0.00	75.2±2.00	89.2±0.65
	KNN	70.3±0.94	72.2±0.95	100.0±0.00	100.0±0.00	93.2±1.94	89.0±1.93	91.5±0.67
	DL	90.2±1.1	94.3±0.99	100.0±0.00	100.0±0.00	100.0±0.00	82.3±1.98	94.5±0.66
Different input relevance: setup 5 made up of 6 classes multiple relationship:	SOM	82.2±1.2	87.1±0.89	80.3±0.98	84.3±0.96	88.2±1.1	87.1±1.40	84.9±0.66
	GNG	88.1±0.38	89.3±0.81	97.2±0.83	88.3±0.77	90.2±0.97	90.7±1.22	90.6±1.16
	DT	94.4±1.12	84.6±0.88	89.6±0.98	95.0±1.26	90.0±1.10	98.0±1.03	91.9±0.69
	LD	97.4±0.76	81.1±0.67	91.1±0.85	91.2±0.81	90.2±0.88	87.1±0.65	89.7±0.69
	NB	96.1±0.45	83.2±0.54	87.3±0.62	88.2±0.91	89.1±0.82	88.2±0.33	88.6±0.36
	SVM	96.2±0.87	86.2±0.76	86.2±0.71	88.3±85	87.4±0.90	89.0±0.89	88.9±0.88
	KNN	91.0±0.64	84.2±0.89	95.5±0.99	95.5±0.94	90.4±0.92	97.2±0.96	92.3±0.93
	DL	90.2±0.88	90.2±0.78	94.0±0.89	93.9±1.12	90.4±1.13	97.4±0.88	92.7±0.98
Setup 6 made up of 6 Classes	SOM	78.2±1.12	78.2±1.89	72.0±0.98	97.0±2.1	78.2±1.1	79.3±1.20	80.5±1.21
	GNG	90.2±0.36	90.3±0.41	89.1±0.73	98.4±0.77	90.3±0.97	90.1±1.12	91.4±0.98
	DT	93.0±1.21	92.0±0.78	91.1±0.98	96.4±1.06	90.3±1.2	91.0±1.30	92.3±1.09
	LD	52.0±1.76	47.0±2.67	50.3±1.85	98.3±2.81	60.1±0.88	61.0±1.15	61.3±0.76
	NB	83.4±0.45	70.0±1.40	71.0±0.52	98.1±0.61	70.8±0.92	76.2±0.33	78.3±0.33
	SVM	56.2±0.57	50.2±0.86	51.1±0.91	97.3±0.65	70.3±0.60	71.2±0.89	65.1±0.58
	KNN	92.0±0.64	93.3±0.80	91.1±0.29	96.2±0.14	94.3±0.09	92.3±0.60	93.0±0.03
	DL	97.3±0.08	98.2±0.07	99.1±0.02	100.0±0.00	100.0±0.00	100.0±0.00	99.2±0.02
Multiple relationship and Different Input Relevance Setup 7 made up of 6 classes	SOM	70.2±1.32	70.3±1.39	72.1±0.78	88.2±0.46	79.3±1.51	70.1±1.48	75.0±0.69
	GNG	80.0±0.48	80.1±0.67	89.0±0.90	87.0±0.77	90.2±0.97	90.0±1.12	86.1±1.05
	DT	72.3±0.95	78.0±0.82	90.2±0.78	65.5±1.60	90.8±0.90	91.21±1.03	81.2±0.65
	LD	52.6±1.53	47.8±0.57	50.2±1.08	80.1±0.89	60.3±0.77	61.2±0.85	58.7±0.95
	NB	74.3±0.47	64.0±0.53	66.1±0.62	85.7±0.71	70.5±0.89	76.8±0.53	55.1±0.56
	SVM	56.2±0.87	50.3±0.36	51.1±0.78	97.3±0.84	70.2±0.97	71.1±0.81	66.0±0.78
	KNN	89.2±0.64	90.4±0.79	83.1±0.59	96.3±0.90	94.3±0.82	80.4±0.66	89.0±1.03
	DL	97.2±0.78	95.3±1.05	88.5±0.89	89.1±1.02	90.2±1.23	90.3±0.88	91.8±1.04

Table 7
Experimental Results for Different Noisy Features.

DATA PROBLEMS WITH DATASETS USED	ALGs	Accuracy % (Mean ± Standard Error)						
	CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	Overall Acc	
Setup 8 comprising 10 Noisy Features with 6 Classes	SOM	78.5±0.11	79.2±0.23	78.0±0.08	77.0±0.40	78.2±0.21	79.4±0.26	78.2±0.66
	GNG	86.1±0.43	86.2±0.33	100.0±0.00	100.0±0.00	88.2±0.06	88.2±0.08	91.5±0.04
	DT	100.0±0.00	100.0±0.00	80.0±0.56	18.2±2.3	98.1±0.02	100.0±0.00	82.7±0.91
	LD	78.9.0±0.07	79.0±0.70	79.5±0.24	78.2±1.10	79.0±0.05	80.0±0.10	79.1±0.27
	NB	73.0±0.00	72.0±0.80	74.4±0.30	73.4±0.90	71.0±0.03	72.0±0.90	72.6±0.53
	SVM	76.0±0.40	75.0±0.06	72.4±0.30	72.2±0.90	74.0±0.50	73.0±0.05	73.1±0.82
	KNN	83.0±0.10	81.0±0.07	83.0±0.38	82.1±0.60	82.0±0.90	82.0±0.60	82.0±0.82
	DL	89.7±0.71	88.0±0.65	100.0±0.00	100.0±0.00	100.0±0.00	100.0±0.00	96.3±0.66
Setup 9 comprising 100 Noisy Features with 6 Classes	SOM	33.1.5±1.12	33.1±1.21	100.0±0.00	100.0±0.00	73.6±1.18	73.6±1.10	68.7±0.76
	GNG	59.8±0.32	59.8±0.34	100.0±0.00	98.3±0.72	89.7±0.93	91.5±1.04	83.2±0.68
	DT	78.0±0.60	77.0±0.82	79.0±0.05	78.0±0.90	80.3±1.11	77.7±1.03	78.5±0.62
	LD	65.2±0.72	65.3±0.63	65.0±0.60	65.0±0.40	65.1±0.82	65.0±0.80	65.3±0.61
	NB	64.1±0.51	66.2±0.51	65.0±0.08	66.0±0.10	65.5±0.82	65.5±0.83	65.1±0.31
	SVM	80.4±0.24	84.1±0.79	76.0±0.08	80.0±0.20	76.1±0.92	84.2±0.81	80.1±0.64
	KNN	81.3±0.64	77.4±0.89	80.2±0.99	78.8±0.94	79.6±0.92	79.4±0.96	79.5±0.93
	DL	87.1±0.73	86.4±0.61	100.0±0.00	98.0±0.09	100.0±0.00	100.0±0.00	95.2±0.48
Setup 10 comprising 1000 Noisy Features with 6 Classes	SOM	54.7±0.12	56.7±0.89	55.0 ±0.98	54.0±0.31	56.0 ±0.21	55.0±0.20	55.6±0.21
	GNG	78.9±0.66	80.2±0.41	80.2±0.63	78.0±0.72	79.1±0.37	79.1±0.12	79.4±0.75
	DT	76.0±0.90	74.7±0.20	70.4±0.43	74.4±1.06	70.2±0.23	74.2±0.37	73.3±0.47
	LD	58.2±0.26	54.0±0.06	57.0±0.83	55.3±0.82	56.8±0.10	58.1±0.15	56.4±0.70
	NB	64.4±0.41	69.0±0.40	68.2±0.52	66.2±0.65	67.4±0.72	68.2±0.34	67.1±0.34
	SVM	72.1±0.52	77.2±0.46	77.0±0.51	77.3±0.65	74.1±0.63	76.2±0.82	75.0±0.51
	KNN	74.0±0.64	76.2±0.80	78.0±0.21	76.1±1.19	78.2±1.01	74.1±0.60	76.1±0.03
	DL	100.0±0.00	100.0±0.00	84.1±0.33	82.2±0.95	100.0±0.00	100.0±0.00	94.4±1.18

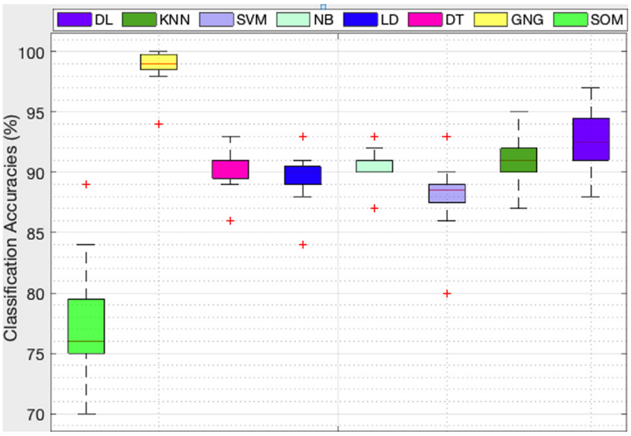

Fig. 1. Classification accuracies for Setup 4: unequally separated classes (6 classes).

expected because, Setup 1 contains no noise and the classes are equally separated. Setup 1 can therefore be thought of as a trivial problem to solve, but one that shows that all methods achieve 100% results which gives a benchmark as the datasets are changed. Also, for Setup 2, apart from DT, all other algorithms had 100% accuracies for classes 3 and 4. However, it is observed that the classification accuracies for classes 1 and 2 are very poor which consequently results in poor overall accuracies for other algorithms except the GNG, which showed the ability to classify datasets where classes are separated with different amounts.

To further investigate and confirm the results from Setup 1 and Setup 2, Setup 3 and Setup 4 were tested on the selected algorithms. Table 6 shows that all the algorithms made 100% classification accuracies for all the 6 classes of the Setup 3 as expected. Similar to the result for Setup 2, all other algorithms except the GNG had poor classification for classes 1, 2 and 6 of Setup 4; the GNG showed a superior performance as further shown in pictorial form in Fig. 1.

5.2. Classes with different input relevance

Dataset made of classes with different input relevance i.e Setup 5 was equally investigated with the selected algorithms. The results in Table 6 for Setup 5 show the performances of the learning algorithm on each of the classes and overall accuracies. The results depict that DL has the best overall accuracy followed by the DT. The reduced classification accuracies of these algorithms confirm their inability to select the relevance features from Setup 5 at class level. This is because, a feature which is relevant for one class can be irrelevant for another class. This makes it necessary for the features selection to be done at class level. Unfortunately, none of the selected algorithms has the ability to carry out feature selection and analysis at class level resulting in the poor accuracies.

5.3. Classes with multiple relationship

In another investigation, the dataset of classes with Multiple relationships (Setup 6) was tested on the learning algorithms. The result is shown in Table 6 for Setup 6. From the result, the DL showed a superior performance as depicted in it having the highest and best classification accuracy amongst the tested classification algorithms. This demonstrates that the DL has the ability to classify datasets with classes made of multiple relationships with negligible classification error.

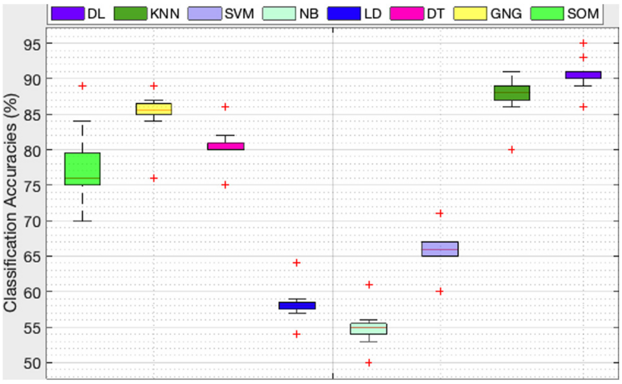

Fig. 2. Classification accuracies for Setup 7: classes with multiple relations and different input relevance.

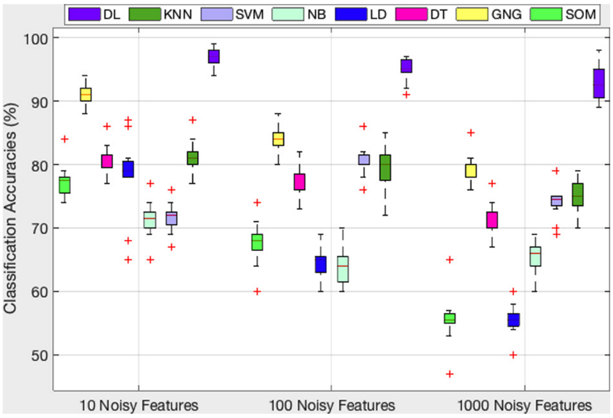

Fig. 3. Experimental results for different noisy features.

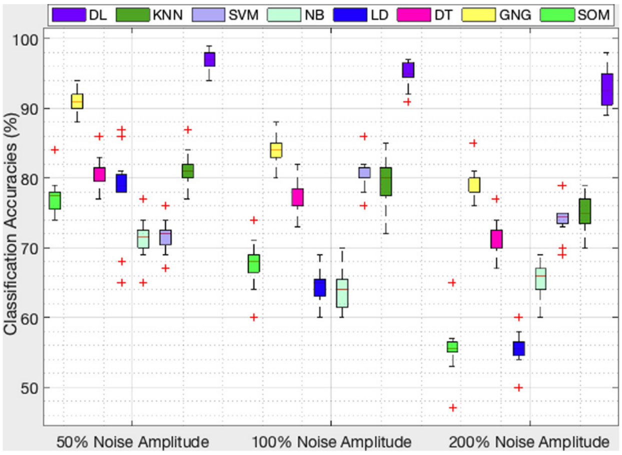

Fig. 4. Experimental results for different noise amplitudes.

5.4. Classes with multiple relationship and different input relevance

Another data problem under study in this paper is the dataset of classes with combined properties of multiple relationships and different input relevance (Setup 7).

The result is shown in Table 6 for Setup 7 and further in Fig. 2. From the result, even though the DL showed the best performance as depicted in it having the highest and best classification accuracy amongst the tested classification algorithms, its accuracy is lower than what was obtained for Setup 6 (dataset with classes made of multiple relationships). This is due to the presence of noisy features with different input relevance which caused the DL to struggle in classification of the data samples, because like other selected algorithms, the DL has unknown ability (since we cannot see what DL has learnt) to carry out feature selection and analysis at the class level.

5.5. Different number of noisy features

We further investigated the performance of the selected algorithms under varying number of noisy features of 10, 100 and 1,000. The results of this experiment is shown in Table 7 and summary in Fig. 3. The results show that other learning algorithms are affected by the increase in number of noisy features. Interestingly, the DL showed the capability of being less affected by number of noisy features.

5.6. Different noise amplitudes

The last data related problem studied in this work is the variations in the amplitude of the noisy features in relation to the maximum amplitude of the information in the dataset. The result of this investigation is shown in Fig. 4. Again, the DL proved that it is also less affected by the amplitude of the noise.

As interesting as this is, it can be observed that the DL, though with the best performance could not achieve 100% classification accuracies on the tested datasets, even though it is possible.

6. Conclusion and future work

In this paper, we have investigated the classification performance of eight learning algorithms on some defined data related problems of interest. The aim of this paper was to investigate the performance of selected classification algorithms on datasets with various properties and problems. The data problems understudied in this research were; datasets with varying inter class distances (classes are separated by different amounts), datasets with classes having different input relevance, datasets with classes defined by multiple relationship, datasets with increasing number of noisy features and datasets with varying amplitudes of noisy features. Also, datasets with combination of some of the problems were also synthesized and tested on the algorithms under consideration. This was in order to mimic the real-world datasets since real world datasets could come with combination of different data related problems, some of which have been identified in the paper. The results of the experimental investigations show that the GNG had the best performance on datasets with varying inter class distances but however performed poorly on datasets with other problems and combination of data related problems.

On the other hand, the DL, performed best on the datasets of different data related problems. As interesting as this is, it was observed that the DL, though with the best performance could not achieve 100% classification accuracies on the tested datasets, even though it is possible. Another major challenge with the DL is its lack of “explainability” as it is a black box.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aggarwal, C. C. (2014). *Data classification: algorithms and applications*. CRC press.

Ahmad, A. U., & Starkey, A. (2018). Application of feature selection methods for automated clustering analysis: a review on synthetic datasets. *Neural Computing and Applications*, *29*(7), 317–328.

Asghar, S., & Iqbal, K. (2009). Automated data mining techniques: A critical literature review. In *2009 international conference on information management and engineering* (pp. 75–79). IEEE.

Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning (vol. 1)*. MIT Press.

Biryulev, C., Yakymiv, Y., & Selemonavichus, A. (2010). Research of artificial neural networks usage in data mining and semantic integration. In *2010 Proceedings of vith international conference on perspective technologies and methods in MEMS design* (pp. 144–149). IEEE.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Boser, B., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, Article 130401.

Bradshaw, J. M., Hoffman, R. R., Woods, D. D., & Johnson, M. (2013). The seven deadly myths of“ autonomous systems”. *IEEE Intelligent Systems*, *28*(3), 54–61.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, *7*(6), 2094–2107.

Chu, G., Lo, P., Ramakrishna, B., Kim, H., Morris, D., Goldin, J., et al. (2014). Bone tumor segmentation on bone scans using context information and random forests. In *International conference on medical image computing and computer-assisted intervention* (pp. 601–608). Springer.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Darbari, A. (2000). *Rule extraction from trained ANN: A survey*. TU Dresden, Germany.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., & Mahoney, M. W. (2007). Feature selection methods for text classification. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 230–239).

Deboeck, G., & Kohonen, T. (2001). *Analiz finansovykh dannykh s pomoshch’yu samoorganizuyushchikhsya kart [Visual Explorations in Finance: With Self-Organizing Maps (Springer Finance)]*. Moscow, Al’pina Publ.

El Kourdi, M., Bensaid, A., & Rachidi, T.-e. (2004). Automatic Arabic document categorization based on the Naïve Bayes algorithm. In *Proceedings of the workshop on computational approaches to arabic script-based languages* (pp. 51–58). Association for Computational Linguistics.

Ezenkwu, C. P., & Starkey, A. (2019). Unsupervised temporospatial neural architecture for sensorimotor map learning. *IEEE Transactions on Cognitive and Developmental Systems*.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Frezza-Buet, H. (2008). Following non-stationary distributions by controlling the vector quantization accuracy of a growing neural gas network. *Neurocomputing*, *71*(7–9), 1191–1202.

Fritzke, B. (1995). A growing neural gas network learns topologies. In *Advances in neural information processing systems* (pp. 625–632).

Fu, Y., Yang, Q., Sun, R., Li, D., Zeng, R., Ling, C. X., et al. (2004). Exploiting the kernel trick to correlate fragment ions for peptide identification via tandem mass spectrometry. *Bioinformatics*, *20*(12), 1948–1954.

Hofmann, A., Schmitz, C., & Sick, B. (2003). Rule extraction from neural networks for intrusion detection in computer networks. In *Conference proceedings. 2003 IEEE international conference on systems, man and cybernetics. conference theme-system security and assurance (Cat. No. 03CH37483) (vol. 2)* (pp. 1259–1265). IEEE.

Hristea, F. T. (2012). *The Naïve Bayes model for unsupervised word sense disambiguation: aspects concerning feature selection*. Springer Science & Business Media.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., & Sundararajan, S. (2008). A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th international conference on machine learning* (pp. 408–415).

Jain, N., & Srivastava, V. (2013). Data mining techniques: a survey paper. *IJRET: International Journal of Research in Engineering and Technology*, *2*(11), 116–119.

Jose, C., Goyal, P., Aggrwal, P., & Varma, M. (2013). Local deep kernel learning for efficient non-linear svm prediction. In *International conference on machine learning* (pp. 486–494).

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th International convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 1200–1205). Ieee.

Kamruzzaman, S., & Sarkar, A. (2011). A new data mining scheme using artificial neural networks. *Sensors*, *11*(5), 4622–4647.

Kim, Y., & Moon, T. (2015). Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, *13*(1), 8–12.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

Kohonen, T., Kaski, S., & Lappalainen, H. (1997). Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. *Neural Computation*, *9*(6), 1321–1344.

Lebbah, M., Rogovschi, N., & Bennani, Y. (2007). BeSOM: Bernoulli on self-organizing map. In *2007 International joint conference on neural networks* (pp. 631–636). IEEE.

Lee, S. J., & Siau, K. (2001). A review of data mining techniques. *Industrial Management & Data Systems*.

Leung, K. M. (2007). k–nearest neighbor algorithm for classification. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*.

Liu, Y., & Huang, H. (2002). Fuzzy support vector machines for pattern recognition and data mining. *International Journal of Fuzzy Systems*, *4*(3), 826–835.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, *17*(4), 491–502.

Meyer, D. (2004). *Support vector machines: The interface to libsvm in package e1071*. Citeseer.

Mishra, A., & Suhas, M. (2016). Classification of benign and malignant bone lesions on CT images using random forest. In *2016 IEEE international conference on recent trends in electronics, information & communication technology* (pp. 1807–1810). IEEE.

Mulak, P., & Talhar, N. (2015). Analysis of distance measures using K-nearest neighbor algorithm on KDD dataset. *International Journal of Science and Research*, *4*(7), 2101–2104.

Ngai, E. W., Xiu, L., & Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, *36*(2), 2592–2602.

Nguyen, M. H., & De la Torre, F. (2010). Optimal feature selection for support vector machines. *Pattern Recognition*, *43*(3), 584–591.

Nguyen, C., Wang, Y., & Nguyen, H. N. (2013). *Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic*. Scientific Research Publishing.

Nikolaou, N. (0000). Reducing dimensionality for face recognition miniproject for machine learning and data mining (COMP61011).

Park, H., Choo, J., Drake, B. L., & Kang, J. (2008). Linear discriminant analysis for data with subcluster structure. In *2008 19th international conference on pattern recognition* (pp. 1–4). IEEE.

Phyu, T. N. (2009). Survey of classification techniques in data mining. In *Proceedings of the international multiconference of engineers and computer scientists (vol. 1)* (pp. 18–20).

Prudent, Y., & Ennaji, A. (2005). An incremental growing neural gas learns topologies. In *Proceedings. 2005 IEEE international joint conference on neural networks, 2005 (vol. 2)* (pp. 1211–1216). IEEE.

Rado, O., Ali, N., Sani, H. M., Idris, A., & Neagu, D. (2019). Performance analysis of feature selection methods for classification of healthcare datasets. In *Intelligent computing-proceedings of the computing conference* (pp. 929–938). Springer.

Rohanizadeh, S. S., & BAMENI, M. M. (2009). A proposed data mining methodology and its application to industrial procedures. *Journal of Optimization in Industrial Engineering (Journal of Industrial*.

Sahu, H., Shrma, S., & Gondhalakar, S. (2011). A brief overview on data mining survey. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, *1*(3), 114–121.

Samuel, A. L. (1988). Some studies in machine learning using the game of checkers. II—recent progress. In *Computer Games I* (pp. 366–400). Springer.

Scharre, P., & Horowitz, M. (2015). *An introduction to autonomy in weapon systems*. Center for a New American Security.

Shi, Z. (2020). Improving k-nearest neighbors algorithm for imbalanced data classification. In *IOP conference series: materials science and engineering (vol. 719)*. (1), IOP Publishing, Article 012072.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.

Steinwart, I., & Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Sui, Y., Zhang, X., Huan, J., & Hong, H. (2019). Exploring data sampling techniques for imbalanced classification problems. In *Fourth international workshop on pattern recognition (vol. 11198)*. International Society for Optics and Photonics, Article 1119813.

Vafaie, H., & Imam, I. F. (1994). Feature selection methods: genetic algorithms vs. greedy-like search. In *Proceedings of the international conference on fuzzy and intelligent control systems (vol. 51)* (p. 28).

Wang, H., Chen, S., Xu, F., & Jin, Y.-Q. (2015). Application of deep-learning algorithms to MSTAR data. In *2015 IEEE international geoscience and remote sensing symposium* (pp. 3743–3745). IEEE.

Westerlund, M. L. (2005). Classification with Kohonen self-organizing maps. *Soft Computing, Haskoli Islands*, *24*.

Witten, I. H., Frank, E., & Hall, M. A. (2005). *Practical machine learning tools and techniques* (p. 578). Morgan Kaufmann.

Xu, S. (2018). Bayesian Naïve Bayes classifiers to text classification. *Journal of Information Science*, *44*(1), 48–59.

Xue, Y., & Hauskrecht, M. (2019). Active learning of multi-class classification models from ordered class sets. In *Proceedings of the AAAI conference on artificial intelligence (vol. 33)* (pp. 5589–5596).

Ye, J., Janardan, R., & Li, Q. (2005). *Two-dimensional linear discriminant analysis, in Advances in Neural Information Processing Systems (vol. 17)*. Cambridge, MA: MIT Press.

Zaffar, M., Hashmani, M. A., & Savita, K. (2017). Performance analysis of feature selection algorithm for educational data mining. In *2017 IEEE conference on big data and analytics* (pp. 7–12). IEEE.

Zhang, K., Lan, L., Wang, Z., & Moerchen, F. (2012). Scaling up kernel svm on limited resources: A low-rank linearization approach. In *Artificial intelligence and statistics* (pp. 1425–1434).

Zhang, M.-L., & Zhou, Z.-H. (2005). A k-nearest neighbor based algorithm for multi-label classification. In *2005 IEEE international conference on granular computing (vol. 2)* (pp. 718–721). IEEE.

Zhu, H., Liu, X., Lu, R., & Li, H. (2016). Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM. *IEEE Journal of Biomedical and Health Informatics*, *21*(3), 838–850.

Uduak Idio Akpan [a,*], Andrew Starkey [b]
[a] *Akwa Ibom State University, Nigeria and University of Aberdeen, UK*
[b] *University of Aberdeen, UK*
E-mail addresses: uduakidio@aksu.edu.ng, u.akpan.18@abdn.ac.uk (U.I. Akpan), a.starkey@abdn.ac.uk (A. Starkey).

[*] Corresponding author.