A Comparative Empirical Analysis of 21 Machine Learning Algorithms for Real-World Applications in Diverse Domains

Isaac Kofi Nti (ntious1@gmail.com)
Sunyani Technical University https://orcid.org/0000-0001-9257-4295
Justice Aning
Sunyani Technical University
Beklisi Ben Kwame Ayawli
Sunyani Technical University
Kyeremeh Frimpong
Sunyani Technical University
Albert Yaw Appiah
Sunyani Technical University
Owusu Nyarko-Boateng
University of Energy and Natural Resources

Research

Keywords: Intrusion detection, heart diseases discovery, financial fraud detection, machine learning algorithms, XGBoost

Posted Date: June 9th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-518365/v1

A Comparative Empirical Analysis of 21 Machine Learning Algorithms for Real-World Applications in Diverse Domains

Isaac Kofi Nti[1*], Justice Aning[2], Beklisi Ben Kwame Ayawli[2], Kyeremeh Frimpong[3], Albert Yaw Appiah[3] and Owusu Nyarko-Boateng[1]

[1]*Department of Computer Science and Informatics, University of Energy and Natural Resources, Sunyani, Ghana*
[2]*Department of Computer Science, Sunyani Technical University, Sunyani, Ghana*
[3]*Department of Electrical & Electronic Engineering, Sunyani Technical University, Sunyani, Ghana*

[*]*Ntious1@gmail.com ; ORCID: [*]0000-0001-9257-4295*

Abstract

The application of computational methods like Machine Learning (ML) and Artificial Intelligence (AI) in several fields has recently received increased attention from academicians and field professionals. However, the availability of several of these computational techniques requires expertise to make the right choice. In making things easy for non-experts and inexperienced practitioners to make a suitable choice, some studies sought to compare the performance of different ML algorithms (MLAs) in solving problems in unique fields. However, most of the previous studies were domain centered on a handful of MLAs, which does not give a fairground for all-inclusive comparative analysis. Also, many of these studies compared MLAs for classification tasks to the best of our knowledge without considering the same MLAs performance for regression problems. This paper examines the performance of twenty-one (21) MLAs for classification and regression tasks based on six datasets from different domains. Empirically we compare their prediction results based on accuracy, balanced accuracy, F-score, Area Under the Curve (AUC), root mean square error, r-squared and adjusted r-squared. The random forest algorithm gave a consistent performance across all the six datasets in classification and regression tasks. However, on average, the XGBoost outperformed all MLAs applied in this study. The dummy algorithm and the linear regression were more moderate than the rest of the applied MLAs in computational complexity. Nevertheless, the overall study outcome shows that MLAs algorithms efficiently solve everyday challenges in elections outcome, financial fraud, network intrusion detection, meteorological forecast and heart diseases discovery; but their performance varies across domains and dataset dimensions.

Keywords: Intrusion detection, heart diseases discovery, financial fraud detection, machine learning algorithms, XGBoost

1. Introduction

Nowadays, technological advancement has led to several opportunities and challenges in different sectors of the economy. Our day-to-day activities have been transformed by technology; from the moment we get up from bed, we begin to mingle with our phones, computers, and vehicles. Fields like finance have seen a glorious door in financial forecasting, mobile banking, e-commerce and e-banking [1], similarly in other fields. However, this advancement has increased credit card fraud, mobile bank frauds and more [2]–[7]. Likewise, the network field is advancing rapidly; several upcoming technologies fundamentally impact how industries and their employees connect. Nevertheless, challenges such as network intrusion, cyber-attacks, phishing, adware, and spyware have increased recently [8]. Also, in the healthcare industry, technology has led to electronic health records (HER), early cancer and heart disease detection; but it has contributed to the rise in the hacking of medical records.

Notwithstanding, one technological advancement that makes the airwave of late is AI, and in particular, its subfield called machine learning. Machine Learning Algorithms (MLAs) have been applied in several domains to help augment human efforts to reduce some of the challenges posed by earlier technologies. Some of these domains include estimating soil cohesion and rainfall-induced soil erosion [9], [10], rainfall and flood forecasting [11]–[14], early heart disease and heart rate detection [15]–[18], financial projection and fraud detection [5], [6], [19], [20], defects in conductors and semiconductors manufacturing [21], and bike-rental estimation [22]. MLAs can be grouped as (i) supervised learning, (ii) semi-supervised learning, (iii) unsupervised learning, and (iv)

reinforcement learning. Of these learning paradigms, many algorithms are available, making it challenging to select the suitable algorithm for a specific domain. Some studies [3], [8], [23]–[26] tried to give a comparative summary of MLAs application in real-world problems to help select the correct one for the right job. Nevertheless, a high percentage of these studies were domain centred. Besides, the supremacy of MLAs as reported in these studies differs, i.e., [27] found neural networks (NN) to outperform support vector machines (SVM) and logistic regression (LR) in predicting rainfall occurrence and intensity. Likewise, [24] found NN to outperform naïve Bayes (NB), K-nearest neighbour (KNN), random forest (RF), SVM and decision trees (DT). However, [9] reported that RF outperformed the SVM and Gaussian regression process (GPR). Also, [4] recorded a better performance with DT than RF, NB, SVM and KNN using imbalanced datasets. These suggest that the performance of machine learning algorithms are partially dependent on the application domain, dataset properties and the skewness of datasets (imbalanced dataset) [1], [4].

Hence, there is a need for comparative performance analysis of several MLAs for classification and regression tasks on different datasets from various domains. Nevertheless, none of the previous studies considered this to the best of our knowledge. This study aims to experimentally assess the performance of twenty-one (21) MLAs on different datasets for classification and regression tasks to determine the algorithms that would improve prediction in elections outcome, credit card fraud detection, network intrusion detection, bike rental estimation, meteorological analysis and heart diseases discovery. Specifically, we examine and compare accuracy metrics, execution times and error metrics of 21 MLAs for classification and regression tasks in six specific domains. Given the previous discussions, this paper seeks to answer the following questions:

Q1: Is there any substantial difference in the performance of machine learning predictive models induced by different algorithms on different datasets from specific domains?

Q2: How does the performance of the same MLAs differ in regression task and classification task on different datasets?

Q3: Which ML among the 21 is the most accurate in classification or regression or both?

The key contributions of this paper are:

1. A preliminary summary of 21machine learning algorithms is presented.
2. We explained the principles of 21 MLAs and their feasibility in several real-life application fields like healthcare, electoral systems, financial fraud, cybersecurity systems, meteorological service and many more.
3. We offer a reference point for industry experts, decision-makers, academicians, and beginners in different real-life conditions and applications areas, mainly from practical viewpoints.

The remaining sections of this paper are organised as follows: Section 2 gives a brief overview of machining learning and a summary of related works. In section 3, we present the methods adopted in this study. Section 4 presents the experimental results and discussions of the study, and section 5 presents the study conclusions.

2. AN OVERVIEW OF MACHINE LEARNING ALGORITHMS

The section presents a brief description of machine learning, ML types, a summary of ML applications in real-world problems and a summary of closely related works.

2.1 Machine Learning

Machine learning algorithms learn with experience and have improved performance in the situations they have already encountered [23]. ML has been applied in several areas like speech recognition, fraud detection, intrusion detection, outlier analysis, etc. Figure 1 shows the basic ML model with input record, where each record, referred to as an instance or example—characterised by a tuple *(x, y)*, where *(x)* is the independent variable and *(y)* dependent feature. Figure 2 shows the primary classifications of MLAs [28].

Figure 1. Basic machine learning model mapping an input attribute set *(x)* into its output *(y)*

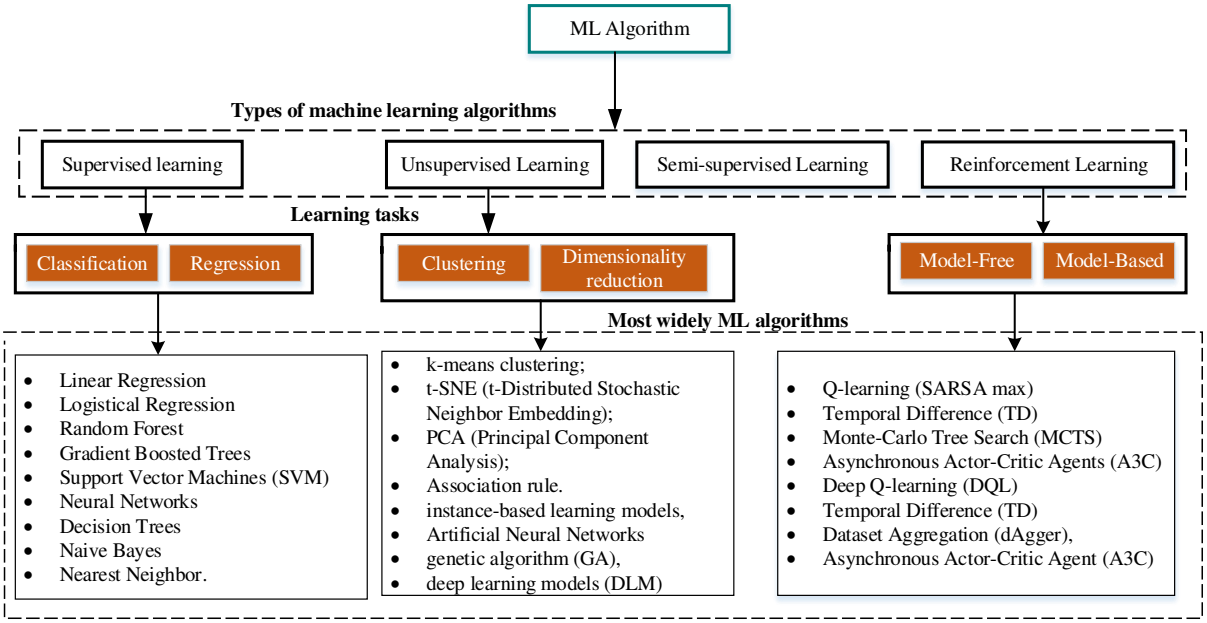

Figure 2: Classification of MLA [28]

2.2 Related works

Researchers and field professionals have tried to propose an efficient MLA for a specific domain problem. In this section, we categorised these studies as follows:

Financial Fraud detection: Lately, financial fraud is increasing rapidly, negatively affecting the economies, administrations, and institutions worldwide [3]. Due to this, several studies have attempted to apply machine learning algorithms to detect various financial frauds. A study carried out by [3] examines three MLA accuracy in detecting credit card fraud. The study reported that Logistic Regression (LR) outperformed Naïve Bayes (NB) and K-Nearest Neighbour (KNN) for detecting fraudulent credit card activities. Similarly, [4] examine the performance of DT, KNN, LR, RF and NB; however, they reported that DT outperformed KNN, LR, RF and NB

Heart disease detection: World Health Organisation (WHO) reports that heart diseases cause 17.9 million deaths each year (World health organisation, 2019). Heart diseases currently lead the chart of fatal diseases; they are considered a prominent cause of death, approximately 31% of all deaths globally. In helping reduce its occurrence, researchers and professionals like the authors of [29] applied different MLA (LR, DT, SVM and RF) to detect early heart diseases. The paper's outcome shows that DT obtained the highest accuracy with some feature engineering than LR, SVM, and RF. However, without feature engineering, RF outperformed all other MLAs. Similarly, a comparative analysis of SVM, LR, Artificial Neural Network (ANN), KNN, NB and DT for accurate heart disease detection in E-Healthcare was presented [16]. The outcome of the paper shows that SVM outperformed all applied MLAs in identifying heart disease.

Network intrusion detection: A significant challenge encountered by the 21st-century network security industry is intrusion detection. Hence, effective monitoring of networks has become paramount. In [8], the study compared nine different MLA (see Table 1) for detecting network intrusion. The study outcome shows the superiority of RF in intrusion detection compared with other MLAs. However, in another study [23], DT was the best for intrusion detection than other state-of-the-art MLAs.

Table 1 shows a summary of related works. From Table 1, previous studies have compared MLAs; however, a higher percentage were focused on unique fields (see Table 1). Additionally, the number of MLAs used by most of these studies is limited (minimal), which does not give a fairground for all-inclusive comparative analysis. Also, as indicated in [1] and [4], the performance of MLAs varies from one field to another due to dataset features. Besides, to the best of our knowledge, many of these studies compared MLA based on the classification task without considering the same MLA performance in the regression task. Therefore, it cannot be generalised based on the outcome of these studies that one MLA is superior to the other. That is, there is the probability that algorithm (*A*) might outperform algorithm (B) in a classification task, yet (B) is better than (A) for regression in the same field or different field. In filling the literature gap, the current study seeks to undertake a comprehensive empirical analysis of twenty-one (21) MLAs for classification and regression tasks, using six (6) datasets cutting across diverse fields.

Table 1. Summary of Related Studies

Ref.	Learning Task	MLA	No. of MLA	Application domain
[8]	CL	BNT, LR, IBK, J48, PART, JRip, RT, RF and REPTree	9	Network Intrusion Detection
[23]	CL	DT, CR, ZeroR, OneR, BNT, NB, NBUpdatable, MLP, SMO, LR, IBI, IBK, Kstar, LWL, J48, decisionstump, AdaboostM1 and inputMappedClassifier	18	Intrusion Detection
[3]	CL	LR, NB and KNN	3	Credit card fraud detection
[4]	CL	DT, KNN, LR, RF and NB	5	Detecting credit card fraud
[29]	CL	SVM, RF, LR		Heart disease identification
[24]	CL	NB, KNN, NN, RF, SVM and DT	6	Druggable proteins detection
[1]	CL/RG	MLP, SVM, DT, AdaBoost, Bagging, Stacking, Blending	7	Stock price prediction
[25]	REG	MLP, SVM, CatBoost, KNN, RF and DT	6	Predict glass transition temperatures
[26]	CL	LR, KNN and LSV	3	Predict lettuce growth stage
[9]	RG	SVM, RF and Gaussian regression process (GPR)	3	Assessment of soil cohesion

CL = Classification; RG = Regression, RF = random forest; RT = random tree, KNN = K-nearest Neighour; NB = Naïve Bayes, BNT = Bayes Net; LR = Logistic regression; DT = decision tree

3. EXPERIMENTAL METHODOLOGY

This section describes the methods used in the ML experiments. We present the study framework; discuss the unique six datasets used in the experimentations and preprocessing methods applied to the dataset. A summary of the MLAs used, accuracy and error metrics adopted in this study are also presented in this section.

3.1 Selected Machine Learning Algorithm

There are several MLAs (see Fig. 2). However, for this study, twenty-one (21) were carefully chosen (see Table 3) based on their popularity and efficiency recorded in a specific domain like Credit card fraud, intrusion detection, elections and heart diseases detection, as reported in the literature [1], [3], [4], [8], [16], [24]. We give a brief description of most of them; however, detailed overviews and tutorials of MLAs exist for readers unfamiliar with their background and use [9], [30]–[35].

1. **Logistic Regression (LR)** is an MLA describing data and explaining the association between one dependent binary feature and two or more ordinal, nominal, interval or ratio-level independent features [29].
2. **Stochastic Gradient Descent (SGD)** is an optimisation technique frequently used in ML applications to find the model hyperparameters that match the best fit between model-predicted values $\left(\hat{y}\right)$ and actual values $\left(y\right)$. It is a simple yet but powerful technique for both regression and classification task.
3. **Multilayer Perceptron (MLP)** is a deep ANN that learns a function $f\left(.\right): R^m \rightarrow R^o$ by training on a dataset (DS), where *(m)* is the dimensions of DS and *(o)* is the output dimensions. It has at least three (3) layers of nodes, namely, (i) the input layer for receiving the input signal, (ii) output layer for making a decision or forecast about the input and (ii) hidden layer(s) sandwich between (i) and (ii) for all computation of the MLP; this layer has an arbitrary number. They are often used for supervised ML tasks. Apart from the input layer, every other node is a neuron that uses a nonlinear activation function [11], [33].
4. **Decision Tree (DT)** is a simple supervised MLA for both regression and classification tasks; however, it is mainly used to classify problems. It is tree-like in construction, where inner nodes characterise the features of a dataset, branches signify the decision rules, and each leaf node signifies the result [1].
5. **Support Vector Machine (SVM)** is a supervised MLA for classification, regression and outliers' challenges. It aims to find an optimal minimal hyperplane *(h)* N-dimensional space (N is the number of features in the dataset) that best divides a given dataset into classes. It is capable of handling both continuous and categorical datasets. The SVM creates an *(h)* in multi-dimensional space to separate different classes by generating optimal (h) in an iterative way, which is used to curtail an error [13]. It gives better accuracy compared with other MLA techniques like LR and DT
6. **Random Forest (RF)** is a supervised MLA commonly used for classification and regression tasks due to its straightforwardness and variability. It is an ensemble learning technique that combines the outcome of two or more decision trees via majority voting or averaging technique depending on the problem at hand, thus, enhancing prediction accuracy [9].
7. **K-Nearest Neighbours (KNN)** is easy to understand and implement supervised MLA for classification or regression tasks. It adopts the resemblance between new data and available data and puts the new data into the most similar group to the available groups. It is a non-parametric algorithm, i.e., it does not make any guess on primary data. It is sometimes referred to as a lazy learner algorithm since it does not learn from the training dataset instantly instead keeps it. At the classification period, it completes an action on the dataset.
8. **Extremely Randomised Trees (ERT),** like RF, is an ensemble MLA that combines the results from many DTs. Its few required hyperparameter and sensible heuristics configuration approaches make it easy to use. It creates a massive number of unpruned DTs based on the training dataset and predicts using majority voting for classification tasks and averaging for regression problems. The primary difference between FR and extra-tress is that it is random in extra-trees during the split on trees and deterministic in RF.
9. **Dummy MLA** is a type of algorithm that adopts simple strategies for prediction without considering the input dataset.
10. **Gradient Boosting Machine (GBM)** put together the estimates from multiple DTs to make the final predictions. It is one of the most potent MLA building predictive models of both regression and classification models. GBM's ensemble shallows DTs sequentially with each DT learning and improving on the earlier one, whereas RF builds an ensemble of deep autonomous DTs [35].
11. **Light Gradient Boosting Algorithm (LightGBM)** is a gradient boosting (GB) framework that employs tree-based learning algorithms. Its nature and design make it faster in training speed and higher efficiency. It can handle vast amounts of a dataset with ease [35].
12. **Extreme Gradient Boosting Algorithm (XGBoost)**, like the LightGBM, is a decision-tree-based ensemble MLA based on the GB framework, i.e., an improvised version of the GBM algorithm. It can be effectively used both for regression and classification ML tasks. The critical difference between RF and GB Machines (GBM) is that while in RF, trees are built independently, GBM adds a new tree to complement already built ones [35].

13. **Adaptive Boosting (AdaBoost)** is seen as the first hands-on boosting algorithm developed in 1996 by Freund and Schapire. It is a boosting method for ensemble ML, where weights are re-assigned to a particular example, with higher weights to wrongly categorised examples [32].
14. **Linear support vector (LSV)** is a form of SVM that has been proven in the literature to guarantee global optimisation for classification or regression tasks in small, medium and large datasets.
15. **Generalised linear model (GLM)** denotes a more important class of models promoted by McCullagh and Nelder (1982, 2nd edition 1989) as cited in [36]. They refer to orthodox LR models for a continuous response variable given categorical or continuous or both forecasters. It comprises multiple linear regression (MLR), plus Analysis of Covariance (ANCOVA) and Analysis of variance (ANOVA).
16. **Bootstrap aggregating (Bagging)** is an ensemble ML designed to expand the solidity and precision of MLAs for both regression and classification. Typically, the bagging techniques are applied to DTs, but they can be effectively used with any other MLA [1]. Its prediction outcome is estimated using the averaging technique. It aids in preventing overfitting of models by decreasing variance.
17. **A passive-aggressive algorithm** is an accessible incremental online learning MLA for regression and classification problems proposed by [37]. Due to its closed-form update rule, its implementation is very straightforward.
18. **Bernoulli Naive Bayes (BernoulliNB)** is part of the Naive Bayes family. It only accepts binary parameters, i.e., BernoulliNB has two commonly exclusive outcomes $P(x=0)=1-p$ or $P(x=1)=p$. It presupposes that all our input features are two-state such that they take only two values
19. **Ridge Regression** is a traditional regularisation method broadly used in ML and statistics. Its natural form is the standard least-squares linear regression, enhanced with an embedded tuneable additive *L2* norm penalty term in the risk function. It sometimes called Tikhonov Regularisation
20. **Nearest Centroid (NC)** is one of the most unappreciated and underutilised MLA, yet, it is moderately powerful and is exceptionally efficient for specific ML tasks. Its operation is similar to that of the KNN.
21. **Linear regression (LeR)** is one of the most famous and used algorithms in ML and statistics. It tries to model the linear association between two variables (*x and y*) by fitting a linear equation to experiential data.

Table 3: Selected MLAs and Hyperparameters Settings

S/N	MLA	Symbol	Hyperparameters
1.	Logistic Regression	LR	penalty= l2, solver='lbfgs', max_iter =100, multi_class='auto', tol=0.0001, C=1.0
2.	Stochastic Gradient Descent	SGD	loss="hinge", penalty="l2", max_iter=5
3.	Multi-Layer Perceptron	MLP	hidden_layer_sizes=100, alpha=0.0001, learning_rate='0.001', activation='relu', solver='adam', batch_size='auto'
4.	Support Vector Machine	SVM	C=1.0, kernel='rbf', degree=3, gamma='scale', *tol=0.001*
5.	Random Forest	RF	n_estimators=150, criterion='gini', max_depth=None, min_samples_split=2
6.	Extreme Gradient Boosting	XGBoost	loss='deviance', learning_rate=0.1, n_estimators=150, subsample=1.0
7.	Extremely Randomized Trees (Extra Trees)	ERT	n_estimators=100, criterion='gini' or ‘mse’, min_samples_split=2
8.	Dummy (Classifier or Regressor)	DUM	strategy='mean’ or ‘prior.’
9.	Adaptive Boosting	AdaBoost	base_estimator=None, n_estimators=50, learning_rate=1.0, loss='linear', algorithm='SAMME.R'
10.	K-Nearest Neighbours	KNN	n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2
11.	Light Gradient Boosting Machine	LGBM	loss='deviance' or 'ls', learning_rate=0.1, n_estimators=150, subsample=1.0
12.	Decision Tree	DT	criterion='gini', splitter='best', max_depth=None, min_samples_split=2,

13.	Linear Support Vector	LSV	penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=1.0, multi_class='ovr'
14.	Generalised linear model	GLM	
15.	Bernoulli Naive Bayes	BNB	alpha=1.0, binarize=0.0
16.	Bootstrap aggregating (Classifier and Regressor)	BAG	n_samples=150, n_features=4, n_targets=1 n_informative=2, n_redundant=0, random_state=0
17.	Passive Aggressive (Classifier and Regressor)	PA	C=1.0, fit_intercept=True, max_iter=1000, tol=0.001, C=1.0, fit_intercept=True, n_iter=5, loss='hinge
18.	Gradient Boosting	GB	loss='deviance', learning_rate=0.1, n_estimators=150
19.	Bayesian Ridge	BR	n_iter=300, tol=0.001, alpha_1=1e-06, alpha_2=1e-06, lambda_1=1e-06
20.	Nearest Centroid	NC	metric='euclidean'
21.	Linear Regression	(LeR)	

3.2 Models Training and Testing Framework

Figure 3 shows the models training and testing framework of this study; we explained each phase in the subsequent section of this paper.

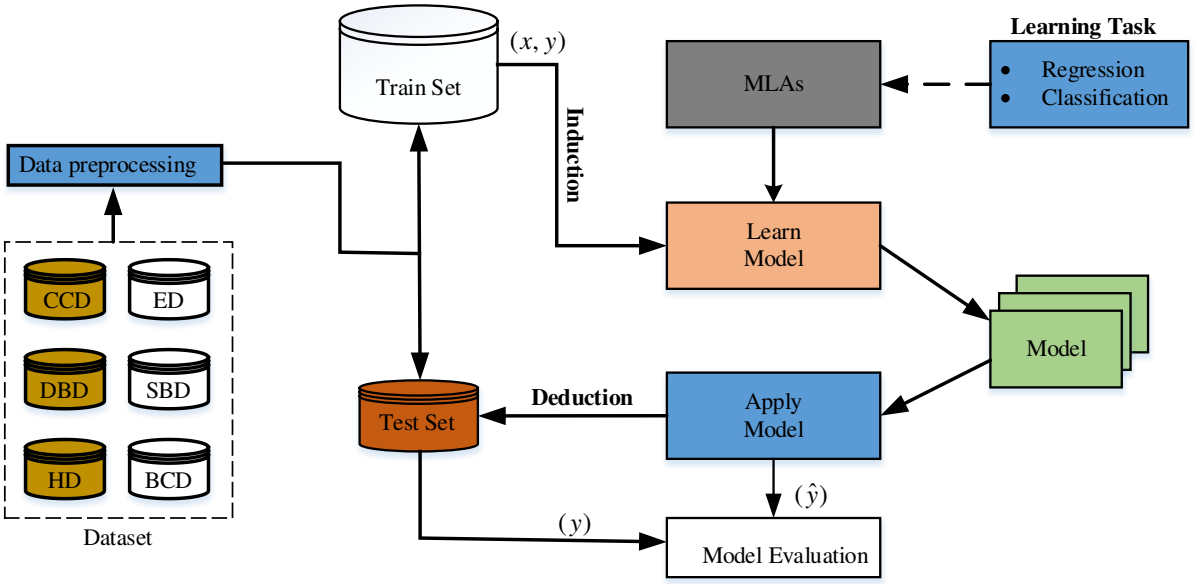

Figure 3: Study Framework

3.2.1 Datasets and Data Preprocessing

Six (6) datasets were downloaded (see Table 2), three for classification task and three for regression task. Each downloaded dataset was preprocessed through (i) data cleaning, i.e., ensuring that the data is correct, reliable and usable by eliminating incorrect, duplicate, corrupted, incorrectly formatted and incomplete data. (ii) Feature scaling, i.e., we standardised the range of features of the datasets within the range of zero (0) and one (1) using min-max normalisation as defined in Eq. (1). We partitioned each clean dataset into two: 80% (train set) and 20% (test set). Some imbalanced datasets were used in this study, allowing us to examine the values of precision and sensitivity for each of the MLAs using the balanced accuracy metric.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

where: x' is the normalisation value; x = the value to be normalised, x_{min} and x_{max} are the minima and maxima value of the dataset.

Table 2: Dataset for Experimental Analysis

S/N	Dataset	Dimension	Task	Source
1.	Credit card fraud dataset (CCD) Credit card transaction covered in European (imbalanced) binary classification	(95162, 31)	CL	K
2.	Heart disease dataset (HD)	(303, 14)		
3.	Dry Bean Dataset (DBD) Dataset of beans species	(13611, 17)		
4.	Election Dataset (ED)	(21643, 27)	RG	A
5.	Bias correction Dataset (BCD) (it contains meteorological forecast data.	(7752, 24)		
6.	Seoul Bike Data (SBD)	(8760, 13)		

K = https://www.kaggle.com; CL = Classification; RG = Regression; *A* = *https://archive.ics.uci.edu*

3.2.2 Models Training

The selected MLAs already discussed in this paper are used to generate a classification model, a regression model, or both. The generated model is then trained (fit) on the training dataset, subsequently applied to the test dataset to make the prediction.

3.2.3 Evaluation Metrics

We compare the performance of the selected MLA using well-known and accepted evaluation metrics. Namely: accuracy, balanced accuracy, AUC, F-Score, Adjusted R-Squared (AdR^2), R-Squared (R^2), and root mean square error (RMSE), as shown in Table 4. We also calculated the balanced accuracy in the classification task to deal with our imbalanced datasets, defined as the average recall on the respective class.

Table 4: Evaluation Metrics

S/N	Technique	Formula	Description
1.	Accuracy (ACC)	$ACC = \left\{ \frac{(TP + TN)}{(TP + FP + TN + FN)} \right\}$	It defines the ratio of all correct predictions to the total number of predictions. TP= true positive, TN = true negative, FN = false negative, FP = false positive
2.	F-Score (FS)	$FS = \frac{2 \times P \times R}{P + R}$	It shows the equilibrium between *(P)* and (R), i.e., is the consonant mean *(P)* and sensitivity (R)
3.	R-squared (R2)	$R^2 = 1 - \frac{RSS}{TSS}$	It shows how much disparity of a dependent feature *(y)* is described by the independent feature(s) RSS = sum of squares of residuals TSS = total sum of squares

4.	RMSE	$RMSE = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\left(y-\hat{y}\right)^2\right)}$	It shows an estimation of the residual between *y* and $\hat{y}$. Where $\hat{y}$ is the model's predicted value, *y* is the true-value, and *n* is the number of observations in test data.
5.	AUC	The AUC is an indicator of the classifier' capability to differentiate between classes, its values fall between 0 and 1. The higher its value the better the performance of the MLAs at differentiating between the positive and negative labels.	

3.3 Experimental Setup

For generalisation purposes, we did not carry out any feature selection or dimensionality reduction in all the six datasets used in this study, i.e., we strictly considered the original features for training and testing the machine learning models. The predictive performance of MLAs is generally affected by assigned values to their hyperparameters. Taking this into thought, in our try-outs, we compared the predictive performance of classification and regression models induced by the twenty-one MLAs with mostly default hyperparameters (see Table 3). We carried out all experiments in this study using the Python programming language with the Scikit-Learn, Matplotlib, pandas and seaborn libraries. A Lenovo (20EGS12E00), Intel® core™ i5-4340M CPU @ 2.90GHz (4 CPUs) 12GB memory was used in this study.

4. RESULTS AND DISCUSSIONS

The outcomes of our experimental results and their implications are discussed in this section.

4.1 Results

We divide the experimental results into two sections, i.e., results based on classification task and regression task.

4.1.1 Classification Task

Figure 4 shows the prediction outcome of the selected classification MLAs for a classification task of detecting credit card fraud. It was observed that XGBoost, ERT, RF obtained the same accuracy (0.9995), making them the most suitable model for accuracy improvement in credit card fraud detection. The XGBoost took 46.3 sec to train, ERT (24.48 sec) and RF (187.37 sec). This result shows that the ERT is computationally moderate than XGBoost and RF yet offers the same accuracy level. Thus, it is suitable to adopt the ERT algorithm for this type of problem for less computational power. The ERT and XGBoost attained the highest F1-score (0.9995), while the RF achieved an F1-score of 0.9994, slightly outperforming the DT (0.991), KNN (0.992), NC (0.991), PA (0.991), LSV (0.991), SVM (0.991) and MLP (0.991) (see Fig. 4). The AUC is a key and vital indicator for measuring the overall performance and the general measure of the accuracy in fraud examination [5]. The higher its value, the better the predictive model performance. The XGBoost obtained an AUC of 0.903, ERT 0.897, RF 0.897. This outcome reveals that the XGBoost is more accurate in predicting credit card fraud than all the elected MLAs in this paper. The compelling results by the XGBoost and ERT in terms of accuracy and AUC suggests that they can effectively be used for studies in this field or use as benchmark algorithms. Surprisingly, the dummy classifier used 0.478 sec to train and gave a prediction accuracy of 0.997, F1-score (0.996) and AUC (0.4991). Also, the PA used 0.895 Sec to train and obtained a prediction accuracy of 0.999, F1-score (0.999) and AUC (0.878). However, there has not been much discussion about these powerful algorithms (shown in figure 4) in the literature. On the other hand, the obtained balance accuracy of the dummy classifier (0.499) shows that it is not suitable for an imbalanced dataset. Though the accuracy of the GLM (0.976) compared with XCBoost, ERT and RF are slightly low, the difference between its balanced accuracy (0.925) compared with that of XCBoost, ERT, and RF tells that the GLM is more suitable for an imbalanced dataset than XCBoost, ERT and RF.

Figure 5 shows the prediction metrics of the selected classification models on the heart disease dataset. Though the KNN, MLP, BernoulliNB and GaussianNB were not among the top 5 algorithms in the credit card detection problem (see fig. 4), they were the top MLAs with the improved accuracy in predicting heart diseases. Logistic regression outperforms SVM, and this supports the findings in [3]. However, KNN outperforms LR, which disagrees with results in [3] for credit card fraud detection. We observed that in predicting heart disease, the XGBoost and ERT are not in the high performing MLAs (top 6), but the RF is within the top 6 in all cases.

Figure 6 shows the performance metrics of models for a classification task on the DBD dataset. It was observed that the XGBoost, LGBM, SVM, LR and RF outperformed all other MLAs in terms of accuracy. Considering the dataset dimensions (see Table 2) and the performance of the XGBoost for the classification task, it suggests that XGBoost is more efficient on a bigger data size. From figures 4-6, the RF has a constancy in performance compared with the remaining MLAs in this study. Thus, the RF is more reliable in all situations, namely financial fraud detecting and heart disease prediction and dry beans species detection. Also, the outcome shows that the RF is consistent with or without data imbalance.

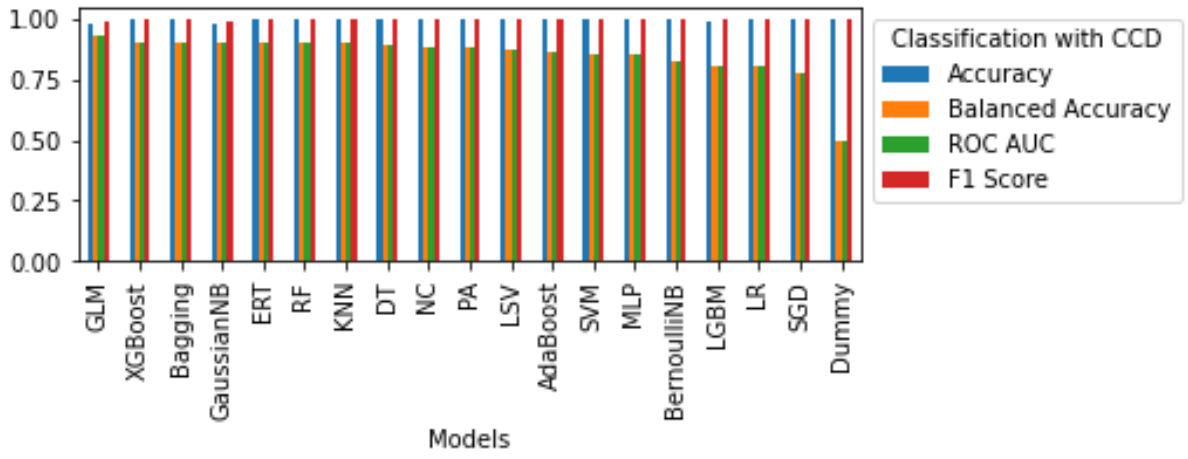

Figure 4: Classification results of models on CCD

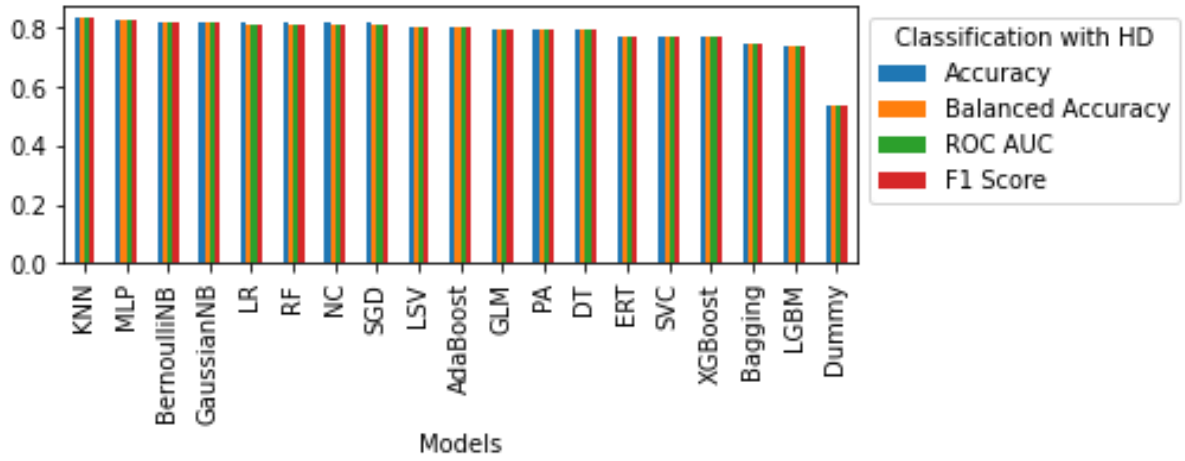

Figure 5: Classification results of models on HD

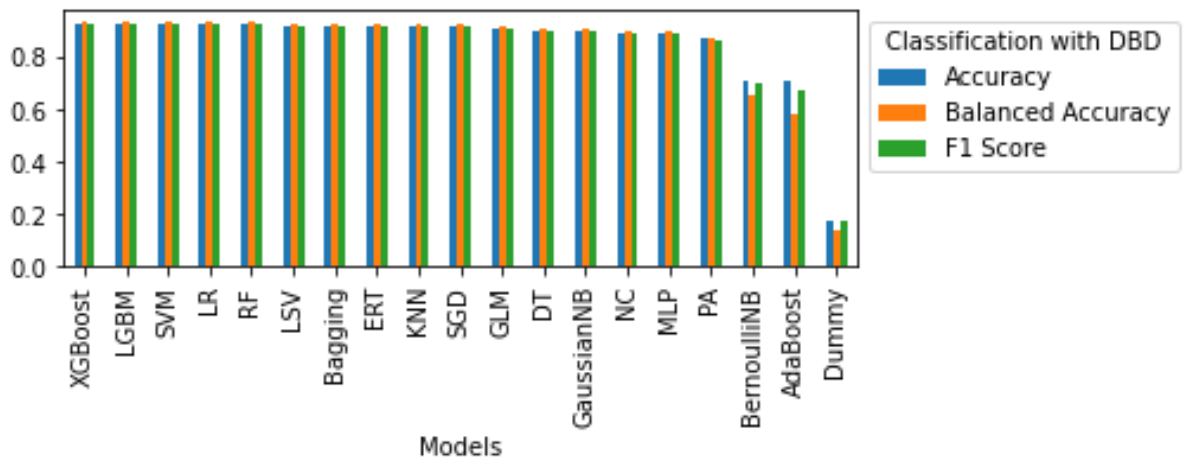

Figure 6: Classification results of models on DBD

Furthermore, the XGBoost performance in these three cases shows that it is pretty not suitable when total observations in the training dataset are significantly lesser than the total number of features. Therefore, we advised it to be used when the training dataset has a large number of observations. The total number of features is minimal compared with the number of records in the training dataset. However, in an attempt to use XGBoost in such an instance, there is a need to adopt an efficient and effective parameter optimisation approach to enhance the predictive performance of the XGBoost.

Table 5 shows the comparison of training time among the selected MLAs for classification tasks on three datasets. The KNN recorded the highest training time (826.9732 sec) on the CCD dataset, followed by the SVM (279.9567 sec). However, in terms of accuracy, the KNN was seventh in the CCD datasets, suggesting that with MLA, a higher training time does not imply higher accuracy. The training time of the XGBoost in all datasets (see Table) proves why it is widely familiar, and many praises from the data science community.

Table 5: Time Taken by Models in Training for Classification on Specific Datasets

	Datasets		
Models	**CCD (sec)**	**DBD (sec)**	**HD (sec)**
Dummy	**0.478709**	**0.027983**	0.014984
NC	0.589662	0.318799	0.019988
BernoulliNB	0.674165	0.079972	0.014989
GaussianNB	0.67861	0.478727	**0.010994**
MLP	0.857507	0.029982	0.115951
PA	0.895487	0.120932	0.114935
SGD	1.03639	1.251298	0.054968
GLM	1.750991	7.181878	0.020008
LR	3.21915	0.031981	0.035978
LGBM	3.678884	0.074954	0.049971
DT	17.97816	0.798557	0.015992
ERT	24.47891	2.801386	0.017972
XGBoost	46.30085	1.97786	0.014975
AdaBoost	53.73608	0.191901	0.014007
LSV	62.83189	0.06896	0.013011
Bagging	107.0303	0.775571	0.011994
RF	187.3735	0.876389	0.162922
SVM	279.9567	0.03498	**0.010993**
KNN	826.9732	1.045382	0.016992

4.1.2 Regression task

Tables 6-8 shows the performance analysis of the selected MLAs for regression tasks on different datasets. We observed that the performance of the ERT is consistent in all three tasks (see Table 6-8). We observed that almost all the tree-based MLAs such as ERT, LGBM, RF, Bagging and XGBoost are within the top 5 MLAs that performed well on all three-regression-based datasets. Therefore, it can be deduced that tree-based algorithms are more efficient for regression tasks. Once again, the consistency of the RF cannot be overlooked (see Tables 6 -8); this shows that the RF is a robust algorithm for both classification and regression tasks. Also, the performance of the ensemble techniques affirms their excellence, as pointed out in the literature [1], [18]. Concerning computational complexity, it can be seen from Tables 6-8 the linear regression (LeR) algorithm is preferable. That is, it recorded the least training time in all the three-regression tasks.

Table 6: Regression results of models on BCD

Models	Adjusted R-Squared	R-Squared	RMSE	Time Taken
ERT	0.928984	0.930014	3.94E-07	2.892325
LGBM	0.924658	0.92575	4.06E-07	0.368182
RF	0.906513	0.907869	4.52E-07	6.030533
Bagging	0.894377	0.895909	4.80E-07	0.726562
KNN	0.88263	0.884332	5.06E-07	0.435748
GB	0.85918	0.861222	5.55E-07	2.222736
LeR	0.798104	0.801032	6.64E-07	0.023666
BR	0.798104	0.801032	6.64E-07	0.037596
SGD	0.794864	0.797839	6.70E-07	0.023985
DT	0.790934	0.793966	6.76E-07	0.116932
AdaBoost	0.75605	0.759587	7.30E-07	0.940169
LSV	0.685354	0.689917	8.29E-07	0.673597
Dummy	-0.01479	-7.27E-05	1.49E-06	0.01899
GLM	-0.01479	-7.27E-05	1.49E-06	0.036269
XGBoost	**-0.01646**	**-0.00172**	**1.49E-06**	**0.204865**
SVM	-0.43882	-0.41795	1.77E-06	0.018989
GaussianProcessRegressor	-1.44533	-1.40987	2.31E-06	8.228861
PA	-16.2857	-16.035	6.15E-06	0.032983
MLP	-5.7E+08	-5.6E+08	0.035236	0.928272

Table 7: Regression results of models on ED

Models	Adjusted R-Squared	R-Squared	RMSE	Time Taken
ERT	0.999994	0.999994	0.000136	2.372637
DT	0.999962	0.999963	0.000351	0.143919
RF	0.999947	0.999947	0.000419	6.91836
Bagging	0.999929	0.999929	0.000484	0.751582
XGBoost	**0.999905**	**0.999906**	**0.000558**	**0.980282**
LGBM	0.999846	0.999847	0.000711	0.486429
GB	0.999339	0.999342	0.001474	3.509719
KNN	0.998919	0.998926	0.001884	1.102067
GaussianProcessRegressor	0.99498	0.99501	0.004061	127.357
AdaBoost	0.9947	0.994731	0.004173	0.421757
LeR	0.992931	0.992973	0.004819	0.055328
BR	0.99293	0.992972	0.004819	0.061965
SGD	0.991226	0.991279	0.005369	0.057965
LSV	0.987119	0.987197	0.006505	3.564305
MLP	0.948834	0.949141	0.012964	2.258847

GLM	0.81679	0.817891	0.024532	0.078379
PA	0.706347	0.708112	0.031058	0.058538
SVM	0.29356	0.297804	0.048172	0.20788
Dummy	-0.00692	-0.00087	0.057512	0.045975

Table 8: Regression results of models on SBD

Models	Adjusted R-Squared	R-Squared	RMSE	Time Taken
LGBM	0.869452	0.870346	0.02271	0.195887
RF	0.863519	0.864454	0.02322	2.626481
ERT	0.860607	0.861562	0.023467	1.751987
XGBoost	0.853226	0.854232	0.02408	0.417759
Bagging	0.850932	0.851953	0.024267	0.295848
GB	0.833749	0.834889	0.025628	0.710606
KNN	0.767582	0.769175	0.030302	0.240864
DT	0.729139	0.730995	0.032712	0.053969
MLP	0.577213	0.58011	0.040869	0.863945
AdaBoost	0.549428	0.552516	0.04219	0.496727
BR	0.532521	0.535725	0.042975	0.019969
LeR	0.532431	0.535636	0.042979	0.01298
SGD	0.531648	0.534858	0.043015	0.018985
LSV	0.480524	0.484084	0.045302	0.404767
GLM	0.445606	0.449405	0.046799	0.022989
SVM	0.385816	0.390025	0.049258	0.092933
PA	0.328305	0.332908	0.051513	0.015971
Dummy	-0.00699	-9.23E-05	0.063073	0.011993
GaussianProcessRegressor	-1.43264	-1.41596	0.098032	11.4715

4.2 Discussion

This study used twenty-one state-of-the-art MLAs to develop classification and regression models for (i) credit card fraud detection, (ii) heart disease detection, (iii) beans species detection, (iv) election results prediction, (v) bike rental prediction, and (vi) meteorological forecast. The performance of each MLA was determined by assessing how correctly their predicted value is to the actual value. The evaluation metrics included the accuracy rate, area under the curve (AUC), F-Score, r-squared, Adjustable r-squared and root-mean-square-error.

Of the twenty-one applied MLAs, RF showed a consistent performance across the dataset in both classification and regression. The RF showed its ability to handle both balanced and imbalanced datasets and smaller and bigger dimensional datasets. The results obtained by the RF in both regression and classification affirms the reports of [9], [29] that RF outperformed the SVM, DT, LR and Gaussian regression process (GPR) in a comparative analysis. The XGBoost outperformed all other MLA in detecting credit card fraud and dry beans species detection. The outcome shows that the popularity of the XGBoost of late among data scientists is in the right direction. However, it was observed that the XGBoost is not suitable where the dataset has more features compared with the number of observations. Nonetheless, we believe that with a suitable optimisation algorithm in tuning its parameters, the XGBoost could perform very accurately in this situation. Notwithstanding, based on the experimental outcome across the six datasets, the XGBoost average performance is better than all MLAs applied in this paper.

Our experimental outcome clearly showed that tree-based MLAs (such as RF, DT, XGBoost, ERT and Bagging) are more accurate in regression tasks than classification problems. More also, they are computationally less expensive compared with their counterparts. The time complexity of MLAs is significant as the accuracy [1]; in this study, it was observed that the dummy algorithm offers a shorter training time compared with the other MLAs

applied in this experimental study. The efficiency of ensemble learning has received more attention lately [1], [18]. The outcome of this paper has affirmed their accuracy; though not high in all cases, it is better than most single MLAs in all cases. The extremely randomised tree (ERT) technique is very accurate in a regression task, even with noisy features.

5. CONCLUSION

The availability of several MLAs makes it challenging to select the best one for the right job. Current studies have compared MLAs, such as DT, RF, LR, SVM and more. Therefore, this paper provided an experimental comparative analysis of twenty-one (21) machine learning algorithms for classification and regression in six domains. We compare their results based on accuracy, recall, precision, F-Score, true-positive rate, true-negative rate, root mean square error and mean absolute error. The study outcome shows that MLAs perform differently on specific domain dataset; thus, one single algorithm is not efficient in all domains. The outcome shows a substantial performance difference in ML predictive models induced by different algorithms on different datasets from specific domains. Also, we observed that about (95%) of the selected MLAs recorded a difference in performance in regression ML task from classification task on different datasets. Therefore, it suggests that an algorithm might be more suitable for a regression task than a classification task. The RF algorithm shows a performance consistency on all six-dataset used in this paper. However, on average, the study outcome shows that the XGBoost algorithm outperformed all other MLAs. However, its performance, where the number of features in a dataset is high compared with the number of observations, was not entirely encouraging.

Nonetheless, the study outcome suggests that the performance of most MLAs is hindered due to the skewness of available datasets (unbalanced). Hence, researchers applying machine learning algorithms on a specific dataset can apply the resampling techniques to the respective datasets being used first. Doing this helps decrease the unevenness ratio of datasets, which might produce better model performance. All 21 MLA used in this study are supervised algorithms; future evaluation of these models with unsupervised MLA is necessary for a more conclusive result. Furthermore, the XGBoost performance with a small dataset can be enhanced further with a suitable optimisation algorithm in future works.

Abbreviation

ML: Machine Learning; AI: Artificial Intelligence; MLAs: Machine Learning Algorithms; AUC: Area Under the Curve; HER: electronic health records; NN: Neural Networks; SVM: Support Vector Machine; LR: Logistic Regression; NB: Naïve Bayes (NB); KNN: K-Nearest Neighbour; RF: Random Forest; DT: Decision Trees; CL: Classification; RG: Regression; BNT: Bayes Net; SGD: Stochastic Gradient Descent; MLP: Multilayer Perceptron; ERT: Extremely Randomised Trees; GBM: Gradient Boosting Machine; LGBM: Light Gradient Boosting Algorithm; XGBoost: Extreme Gradient Boosting Algorithm; AdaBoost: Adaptive Boosting; LSV: Linear Support Vector; GLM: Generalised Linear Model; MLR: Multiple Linear Regression; ANCOVA: Analysis of Covariance; ANOVA: Analysis of Variance; NC: Nearest Centroid; LeR: Linear Regression; BNB: Bernoulli Naive Bayes; BAG: Bootstrap aggregating; PA: Passive Aggressive; BR: Bayesian Ridge; CCD: Credit card fraud dataset; HD: Heart disease dataset; DBD: Dry Bean Dataset; ED: Election Dataset; BCD: Bias correction Dataset; SBD: Seoul Bike Data; AdR^2: Adjusted R-Squared; R^2: R-Squared; RMSE: Root Mean Square Error.

Acknowledgements: Not applicable.

Availability of data and materials: The datasets used and/or analysed during the current study are publicly available.

Conflict of interest: Not applicable.

Funding: Authors did not receive any funding for this study.

Authors' contributions: IKN obtained the datasets for the research and performed the initial experiments. IKN, JA, BBKA, KF, AYA and ONB contributed to the manuscript development modification of study objectives and methodology. All authors contributed to the editing and proofreading. All authors read and approved the final manuscript.

Ethics approval and consent to participate: Not applicable.

Consent for publication: Not applicable.

REFERENCE

[1] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A comprehensive evaluation of ensemble learning for stock-market prediction," *J. Big Data*, vol. 7, no. 1, p. 20, Dec. 2020, doi: 10.1186/s40537-020-00299-5.

[2] A. Singh and A. Jain, "An Empirical Study of AML Approach for Credit Card Fraud Detection–Financial Transactions," *Int. J. Comput. Commun. Control*, vol. 14, no. 6, p. 670, Feb. 2020, doi: 10.15837/ijccc.2019.6.3498.

[3] F. Itoo, Meenakshi, and S. Singh, "Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection," *Int. J. Inf. Technol.*, Feb. 2020, doi: 10.1007/s41870-020-00430-y.

[4] S. Khatri, A. Arora, and A. P. Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2020, pp. 680–683, doi: 10.1109/Confluence47617.2020.9057851.

[5] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020, doi: 10.1109/ACCESS.2020.2971354.

[6] H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah, and Q. Kang, "Optimizing Weighted Extreme Learning Machines for imbalanced classification and application to credit card fraud detection," *Neurocomputing*, vol. 407, pp. 50–62, Sep. 2020, doi: 10.1016/j.neucom.2020.04.078.

[7] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, "Spatio-Temporal Attention-Based Neural Network for Credit Card Fraud Detection," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, pp. 362–369, Apr. 2020, doi: 10.1609/aaai.v34i01.5371.

[8] S. Choudhury and A. Bhowal, "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, May 2015, no. May, pp. 89–95, doi: 10.1109/ICSTM.2015.7225395.

[9] L. Hai-Bang, T.-A. Nguyen, and B. T. Pham, "Estimation of Soil Cohesion Using Machine Learning Method: A Random Forest Approach," *Adv. Civ. Eng.*, vol. 2021, no. Ml, pp. 1–14, Mar. 2021, doi: 10.1155/2021/8873993.

[10] T. V. Dinh, H. Nguyen, X.-L. Tran, and N.-D. Hoang, "Predicting Rainfall-Induced Soil Erosion Based on a Hybridization of Adaptive Differential Evolution and Support Vector Machine Classification," *Math. Probl. Eng.*, vol. 2021, pp. 1–20, Feb. 2021, doi: 10.1155/2021/6647829.

[11] H. Abdel-Kader, M. A.-E. Salam, and ..., "Hybrid Machine Learning Model for Rainfall Forecasting," *J. Intell. ...*, vol. 1, no. 1, pp. 5–12, 2021, doi: 10.5281/zenodo.3376685.

[12] Y. Tikhamarine *et al.*, "Rainfall-runoff modelling using improved machine learning methods: Harris hawks optimizer vs. particle swarm optimization," *J. Hydrol.*, vol. 589, no. March, p. 125133, Oct. 2020, doi: 10.1016/j.jhydrol.2020.125133.

[13] C. Z. Basha, N. Bhavana, P. Bhavya, and S. V, "Rainfall Prediction using Machine Learning & Deep Learning Techniques," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Jul. 2020, no. Icesc, pp. 92–97, doi: 10.1109/ICESC48915.2020.9155896.

[14] D. S. Rani, G. N. Jayalakshmi, and V. P. Baligar, "Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Mar. 2020, no. Icimia, pp. 261–267, doi: 10.1109/ICIMIA48430.2020.9074928.

[15] P. Sharma, K. Choudhary, K. Gupta, R. Chawla, D. Gupta, and A. Sharma, "Artificial plant optimization algorithm to detect heart rate & presence of heart disease using machine learning," *Artif. Intell. Med.*, vol. 102, p. 101752, Jan. 2020, doi: 10.1016/j.artmed.2019.101752.

[16] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, and A. Saboor, "Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare," *IEEE Access*, vol. 8, no. Ml, pp. 107562–107582, 2020, doi: 10.1109/ACCESS.2020.3001149.

[17] I. Tougui, A. Jilbab, and J. El Mhamdi, "Heart disease classification using data mining tools and machine

learning techniques," *Health Technol. (Berl).*, vol. 10, no. 5, pp. 1137–1144, Sep. 2020, doi: 10.1007/s12553-020-00438-1.

[18] B. A. Tama, S. Im, and S. Lee, "Improving an Intelligent Detection System for Coronary Heart Disease Using a Two-Tier Classifier Ensemble," *Biomed Res. Int.*, vol. 2020, pp. 1–10, Apr. 2020, doi: 10.1155/2020/9816142.

[19] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction," *J. Big Data*, vol. 8, no. 1, p. 17, Dec. 2021, doi: 10.1186/s40537-020-00400-y.

[20] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "Predicting Stock Market Price Movement Using Sentiment Analysis: Evidence From Ghana," *Appl. Comput. Syst.*, vol. 25, no. 1, pp. 33–42, May 2020, doi: 10.2478/acss-2020-0004.

[21] B. Roter and S. V. Dordevic, "Predicting new superconductors and their critical temperatures using machine learning," *Phys. C Supercond. its Appl.*, vol. 575, no. May, p. 1353689, Aug. 2020, doi: 10.1016/j.physc.2020.1353689.

[22] S. V. E, J. Park, and Y. Cho, "Using data mining techniques for bike sharing demand prediction in metropolitan city," *Comput. Commun.*, vol. 153, no. December 2019, pp. 353–366, Mar. 2020, doi: 10.1016/j.comcom.2020.02.007.

[23] Y. Hamid, M. Sugumaran, and L. Journaux, "Machine Learning Techniques for Intrusion Detection," in *Proceedings of the International Conference on Informatics and Analytics*, Aug. 2016, vol. 25-26-Augu, pp. 1–6, doi: 10.1145/2980258.2980378.

[24] A. A. Jamali, R. Ferdousi, S. Razzaghi, J. Li, R. Safdari, and E. Ebrahimie, "DrugMiner: comparative analysis of machine learning algorithms for prediction of potential druggable proteins," *Drug Discov. Today*, vol. 21, no. 5, pp. 718–724, May 2016, doi: 10.1016/j.drudis.2016.01.007.

[25] E. Alcobaça *et al.*, "Explainable Machine Learning Algorithms For Predicting Glass Transition Temperatures," *Acta Mater.*, vol. 188, pp. 92–100, Apr. 2020, doi: 10.1016/j.actamat.2020.01.047.

[26] S. C. Lauguico, R. S. Concepcion II, J. D. Alejandrino, R. R. Tobias, D. D. Macasaet, and E. P. Dadios, "A Comparative Analysis of Machine Learning Algorithms Modeled from Machine Vision-Based Lettuce Growth Stage Classification in Smart Aquaponics," *Int. J. Environ. Sci. Dev.*, vol. 11, no. 9, pp. 442–449, 2020, doi: 10.18178/ijesd.2020.11.9.1288.

[27] J. Diez-Sierra and M. del Jesus, "Long-term rainfall prediction using atmospheric synoptic patterns in semi-arid climates with statistical and machine learning methods," *J. Hydrol.*, vol. 586, no. March, p. 124789, Jul. 2020, doi: 10.1016/j.jhydrol.2020.124789.

[28] I. K. Nti, A. F. Adekoya, B. A. Weyori, and O. Nyarko-Boateng, "Applications of artificial intelligence in engineering and manufacturing: a systematic review," *J. Intell. Manuf.*, Apr. 2021, doi: 10.1007/s10845-021-01771-6.

[29] H. Ahmed, E. M. G. Younis, A. Hendawi, and A. A. Ali, "Heart disease identification from patients' social posts, machine learning solution on Spark," *Futur. Gener. Comput. Syst.*, vol. 111, pp. 714–722, Oct. 2020, doi: 10.1016/j.future.2019.09.056.

[30] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "AN OVERVIEW OF MACHINE LEARNING," in *Machine Learning*, Elsevier, 1983, pp. 3–23.

[31] G. Bonaccorso, *Machine learning algorithms*. Packt Publishing Ltd., 2017.

[32] A. Dey, "Machine Learning Algorithms: A Review," *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 3, pp. 1174–1179, 2016, [Online]. Available: www.ijcsit.com.

[33] D. H. Huson, R. Rupp, and C. Scornavacca, "Algorithms and applications," in *Phylogenetic Networks*, vol. 7, no. 13, Cambridge: Cambridge University Press, 2011, pp. 185–186.

[34] S. Abirami and P. Chitra, "Energy-efficient edge based real-time healthcare support system," in *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*, P. Raj and E. Preetha, Eds. 2020, pp. 339–368.

[35] B. Boehmke and B. M. Greenwell, *Hands-on machine learning with R*. CRC Press, 2019.

[36] R. Wolfinger and M. O'connell, "Generalized linear mixed models a pseudo-likelihood approach," *J. Stat. Comput. Simul.*, vol. 48, no. 3–4, pp. 233–243, Dec. 1993, doi: 10.1080/00949659308811554.

[37] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online Passive-Aggressive Algorithms Koby," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, 2006, [Online]. Available: https://u.cs.biu.ac.il/~jkeshet/papers/CrammerDeKeShSi06.pdf.