Analyzing Predictive Algorithms in Data Mining for Cardiovascular Disease using WEKA Tool

Aman, Rajender Singh Chhillar
Department of Computer Science and Applications
Maharshi Dayanand University
Rohtak, India

***Abstract*—Cardiovascular Disease (CVD) is the foremost cause of death worldwide that generates a high percentage of Electronic Health Records (EHRs). Analyzing these complex patterns from EHRs is a tedious process. To address this problem, Medical Institutions requires effective Predictive Algorithms for the Prognosis and Diagnosis of the Patients. Under this work, the current state-of-the-art studied to identify leading Predictive Algorithms. Further, these algorithms namely Support Vector Machine (SVM), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN), Logistic Regression (LR), AdaBoost and k-Nearest Neighbors (k-NN) analyzed against the two datasets on open-source WEKA software. This work used two similar structured datasets i.e., Statlog Dataset and Cleveland Dataset. For Pre-Processing of Datasets, The missing values were replaced with the Mean value and later 10 Fold Cross-Validation was utilized for the evaluation. The result of the performance analysis showed that SVM outperforms other algorithms against both datasets. SVM showed an accuracy of 84.156% against the Cleveland dataset and 84.074% against the Statlog dataset. LR showed a ROC Area of 0.9 against both datasets. The findings of the work will help Health Institutions to understand the importance and usage of Predictive Algorithms for the automatic prediction of CVD based on the symptoms.**

Keywords*—Logistic regression (LR); support vector machine (SVM); Statlog; Cleveland; WEKA

I. Introduction

The heart is a vital organ that circulates rich oxygenated blood through coronary arteries. When these arteries block, such a situation is term as CVD. Major risk factors mostly relate to the patient's lifestyle (e.g., eating behaviour, obesity, smoking, alcohol, and physical inactivity). Global Burden of Disease (2019) reported that nearly a quarter of all deaths in India is because of CVD [1]. It is estimated that every year average of 17 million people dies from CVD, reported by World Health Organization (2019) [2]. There is another report in which the Lancet Medical Journal (2019) reported that women in India are more vulnerable than men [3]. Analyzing complex and similar EHRs is not a cost and effort effective solution.

Predictive algorithms in Data Mining have been used for finding patterns and generalize this for prediction in the last few decades. In our previous work [4], we have discussed: (1) The state-of-the-art for the usage of Data Mining in the Health Sector, (2) Top ten causes of Deaths from chronic Disease. One of the foremost applications of Data Mining in the Health Sector is to build an effective Clinical Disease Prediction System (CDPS) by the algorithm(s). Poorly designed CDPS can be devastating and may result in unwanted outcomes. But properly designed and analyzed CDPS will help hospitals to reduce their expenses. Traditional decision making in healthcare facilities is heavily reliant on the instincts and skills of doctors, rather than the amount of data concealed in EHRs. The consequences of this will be unintentional biases, mistakes, and superfluous medical costs that will impact patient care.

Before analyzing the algorithms, we had several questions like what algorithms to choose for CVD prediction, and on what basis. So, we put them as Research Questions (RQ) and later analyzed them on WEKA Tool. RQ for unbiased and effective analysis of algorithms are as follow:

- RQ1: What are the leading algorithms for the prediction of CVD after extensive study of related work?
- RQ2: Out of these, which Algorithm(s) outperforms other algorithms in terms of performance analysis?

This work divides into multiple Sections. Section II discusses related work by various researchers related to the prediction of CVD using data mining algorithms. Section III outlines the Methodology for performance analysis of the algorithms. This section briefly discusses the datasets, performance metrics, Software, and leading predictive algorithms. Section IV discusses the result of the analysis.

II. Related Work

To answer the RQ1, we have collected several research papers related to CVD from various sources such as IEEE Xplore, Google Scholar, Scopus, and Springer. Then these papers were filtered out based on the Year of Publication (2019—2021). This will help to find the recent usage of algorithms in the prediction of CVD. After extensive study, we have compiled the list of popular algorithms in Table I that answer the first research question. Further, these algorithms will use for performance analysis.

Muniasamy *et al.* [5] stressed on usage and applications of Machine Learning (ML) techniques for CVD prediction. They have used six algorithms viz. SVM, DT, k-NN, RF, and Linear Discriminant Analysis (LDA), Multilayer perceptron (MLP/ANN). They have used four heart datasets (i.e., Cleveland, Switzerland, Hungary, Long Beach VA) available on the UCI (University of California, Irvine) repository. They

used 10-fold cross-validation for splitting training and testing data on WEKA software. Later their performance was evaluated using Metrics. Their work concluded that four algorithms i.e., LDA, RF, DT, and MLP suitable for the prediction of CVD.

Deshmukh et al. [6] suggested a Heart Disorder Prognosis System, in which they used two datasets from the UCI ML repository (i.e., Hungary, Cleveland dataset). They applied k-NN, ANN, DT, and SVM on described datasets using Python Programming language. Their result concluded that DT/ID3 outperform other algorithms on both datasets with the accuracy of 84.08% and 100%, respectively.

Garg et al. [7] performed a comparative analysis of five Data Mining Algorithms namely k-NN, NB, RF, SVM on four datasets collected from the UCI repository (i.e., Cleveland, Switzerland, Hungary, Long Beach VA). The analysis was performed using Python Programming language and concluded SVM outperforms others in terms of accuracy.

Katarya & Meena [8] used the python programming language to study the advantages and disadvantages of eight algorithms viz. LR, NB, SVM, k-NN, DT, RF, ANN/MLP, Deep Neural Network (DNN) for prediction of CVD.

Karun [9] performed a comparative analysis to find the best suitable model for the Prediction of CVD. They used the heart disease dataset from the UCI repository and concluded that RF outperforms other algorithms i.e., SVC/SVM, and k-NN.

Li et al. [10] proposed a feature selection algorithm i.e., "Fast Conditional Mutual Information (FCMIM)". In their work, the Cleveland heart disease dataset was used, collected from the UCI repository. During pre-processing of data, data normalized by min-max scalar and then visualized using heat-map to understand the correlation. In the next phase, feature selection techniques viz. LASSO, MRMR, Relief, and FCMIM were applied to extract relevant features out of the dataset. To check the performance of each feature selection, data was passed to various classifiers (i.e., DT, ANN, LR, k-NN, SVM, and NB). Research work concluded that FCMIM when used with an SVM classifier gives better accuracy and reduces execution time than other cases.

Singh & Kumar [11] calculated the accuracy of various heart prediction algorithms such as SVM, k-NN, and Linear Regression classifiers. This work utilized the heart disease dataset from the UCI repository and then split it into 73% as a training dataset, 37% as a testing dataset. During the pre-processing phase, data balancing and feature selection were carried out on Jupyter (Python). Research work concluded that k-NN perform better than other classifiers in terms of accuracy (87%).

Choudhary & Narayan Singh [12] suggested using AdaBoost over DT because DT may lead to the over-fitting problem. They used the Cleveland dataset and experimented with the python programming language. Results concluded that AdaBoost gives almost the same accuracy (89%) at test sizes 40% and 10% of the model.

Sangle et al. [13] analyzed the theoretical aspect of different work in the field of ML and Deep Learning (DL) for the prediction of Cardiovascular Disease. They have studied the pros/cons of techniques like DT, k-NN, SVM, NB, ANN, and Ensemble Learning. Finally, the authors suggested using ensemble learning/hybrid models to boost the CVD model's prediction accuracy. Shah et al. [14] discussed and experimented with various predictive algorithms like NB, k-NN, DT, and RF where k-NN outperform other algorithms at k=7 in terms of accuracy. They have used the Cleveland dataset and analyzed it with Python Programming language.

Peng et al. [15] presented and discussed the importance/usage of ANN in the prediction of Cardiovascular disease. They have discussed previous work by various researchers related to neural networks for the prediction of CVD.

Hamdaoui et al. [16] proposed a clinical predictive system for Cardiovascular disease. They have used various algorithms like NB, k-NN, SVM, RF, and DT and then applied them to the Cleveland dataset. They used two separate validation techniques i.e., 10-Fold cross-validation, and 70-30 Split validation. In both, the scenario NB outperforms other algorithms. In Split validation, NB gets higher accuracy (84.28%) than Cross-Validation (82.17%).

Kumar et al. [17] calculated various performance metrics like Accuracy, AUC ROC score, and execution time of various classifiers such as RF, DT, LR, SVM, and k-NN. It utilizes a heart disease dataset from the UCI repository and was carried out on Jupyter (Python). Research work concluded that RF performs better in terms of accuracy (85%), ROC AUC score (0.8675), and execution time (1.09 sec).

Santhana Krishnan & Geetha [18] concluded that DT (accuracy=91%) perform better than NB (accuracy=87%) in terms of handling heart medical dataset. The experiment was carried out using Python Programming language by utilizing the heart disease dataset from the UCI repository.

Mohan et al. [19] presented a hybrid CVD prediction model based on RF with a Linear model. Feature selection was carried out using DT entropy and then the result passed to various classifiers like NB, Linear Model, LR, Deep Learning, DT, RF, Gradient Boost Trees, SVM, VOTE, and proposed model HRFLM. An experiment was carried out on R Studio and the result concluded that HRFLM produced better accuracy (88.47%) than other classifiers.

TABLE I. LIST OF ALGORITHMS WITH THEIR REFERENCE COUNT USED IN RELATED WORK

Algorithm	References	Count
SVM	[5], [6], [17], [19], [7]–[11], [13], [14], [16]	11
NB	[7], [8], [10], [11], [13], [14], [16], [18], [20]	9
DT	[5], [6], [18], [19], [8], [10]–[14], [16], [17]	12
k-NN	[5]–[10], [13], [14], [16], [17]	10
LR	[8], [10], [17], [19]	4
ANN	[5], [6], [8], [10], [13]–[15], [19], [20]	9
RF	[5], [7]–[9], [16], [17], [19]	7
Boosting	[12], [19]	2
LDA (others)	[5]	1

Repaka et al. [20] developed Smart heart Disease Prediction (SHDP) that collect heart-related data of the users and predict risk. AES (Advanced Encryption Standard) was used while storing the data, which helps in increasing data security. The research concluded that NB performs better than SMO (Sequential Minimal Optimization), Bayes Net, and MLP regarding accuracy and execution time.

III. Methodology

To answer the RQ2, This work purposed a methodology for finding which algorithm outperforms other algorithms in terms of performance. The complete and step-by-step workflow has shown in Fig. 1. This Section divides into four sections: (1) Datasets used and their pre-processing, (2) Algorithms selected from the first research question, (3) Software used for analysis, (4) Performance metrics.

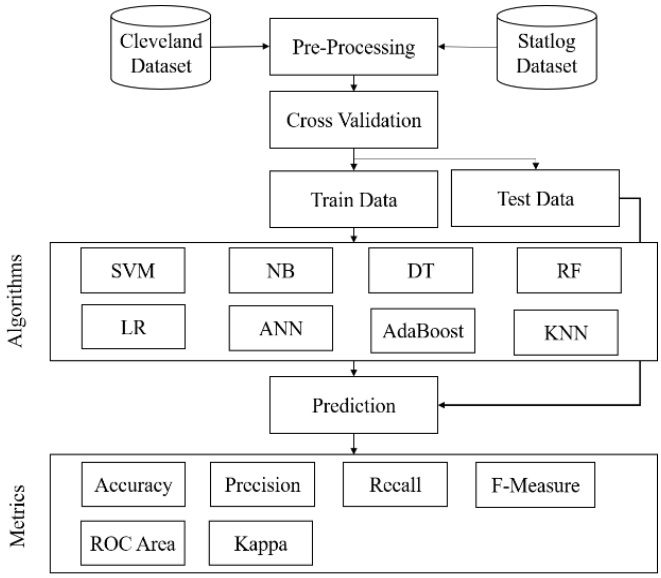

Fig. 1. Methodology for CVD Prediction.

A. Datasets

We have used two similar structured datasets related to CVD (i.e., Cleveland Dataset, Statlog Dataset). Both of these were collected from UCI ML Repository [21][22] and their properties in mentioned in Table II. Cleveland dataset contains 76 attributes, but only 14 attributes are usable for CVD prediction. In this dataset Age, Tresbps, Chol, Thalach, Oldpeak, and Ca are of numeric type and others are of Nominal type. Statlog dataset has 13 feature attributes. Unlike Cleveland dataset, it does not have any missing values. The goal of these datasets is to predict whether the patient is may suffer from CVD in the future or not based on feature attributes. If the outcome of the target variable comes Yes then it means the presence of Cardiac disease else not.

TABLE II. Properties of Datasets

Properties	Cleveland Dataset	Statlog Dataset
Number of Attributes	14	14
Number of Instances	303	270
Missing Values	Yes	No

B. Selected Algorithms

The selection of algorithm(s) largely depends on the Dataset and type of problem (e.g., classification, clustering etc.). Table I shows the list of popular algorithms after the extensive study (RQ1). In this sub-section, algorithms that had $Research\ Count \geq 2$ in Table I is discussed.

1) Support vector machine: SVM identifies the hyperplane with the greatest distance between two classes (see Fig. 2) [23]. The supporting vectors are the vectors (cases) forming the hyperplane. Researchers/Scholars must optimize the distance between hyperplanes. SVM employs a non-linear kernel function to map information at a place where a linear hyperplane cannot isolate the data. The kernel trick is the kernel function, which converts the data into a higher dimensionality, allowing for linear separation. In this work, we have used SMO (Sequential Minimal Optimization) function in the WEKA tool.

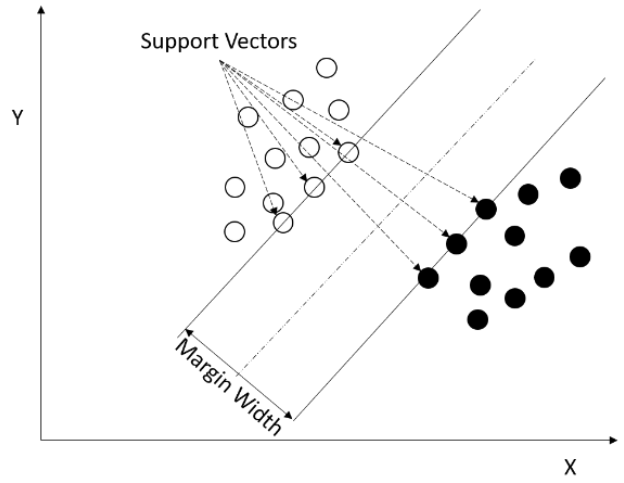

Fig. 2. Linear Support Vector Machine's Architecture.

2) Naïve Bayes: The foundation of the NB classifier is grounded on the theorem of Bayes (see Equation (1)) with the assumptions of independence among predictors [24]. An iterative parameter estimate that is especially useful for the very largest datasets is simple to construct, without a complicated iteration model. NB classifier does not struggle to be very simple and often works extremely well, as it often beats more complex classification methods. Here, we have used the NaiveBayes filter in the WEKA tool.

$$P(K|L) = \frac{P(L|K) \times P(K)}{P(L)} \quad (1)$$

Where $P(K|L)$ is the possibility of occurrence of K if L has already happened; $P(L|K)$ is the possibility of occurrence of L if K has already happened; $P(K)$, $P(L)$ is the independent possibility of event K and L respectively.

3) Decision tree: DT builds a prediction model in the shape of a tree structure [25]. DT provides a simple graphical solution to the problem which makes it most easily understandable among all classifiers. DT divides a dataset into

successively smaller subgroups while building a new decision tree. The end output is a tree with decision/prediction and leaf nodes. The decision node has two or more branches (for example obesity? exercise?). A classified node (e.g., Unfit, Fit) is a decision as shown in Fig. 3. If Age < 40 and the person is Obese then it means the Patient is Unfit. If Age > 40 and not doing Exercise then the patient is unfit. DT is capable of handling both numerical and nominal/categorical attribute types. We have used the J48 (Implementation of DT based on JAVA) function in WEKA Tool.

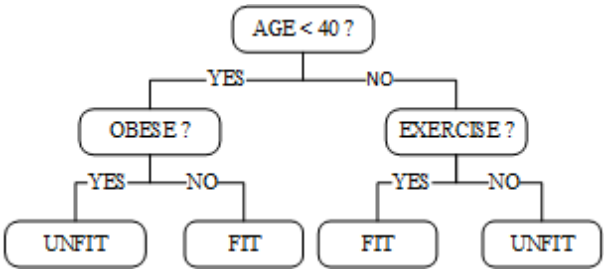

Fig. 3. Decision Tree for Obesity.

4) Random forest: RF (i.e., Random Forest) is a classifier that advances from DTs as shown in Fig. 4 and it consists of many decision trees [26]. Each decision tree provides training data as input and then their result aggregates and most voted will result as a prediction. Overfitting is a common concern in DT; RF aids in preventing this problem. Here, we have used the RandomForest function in WEKA Tool.

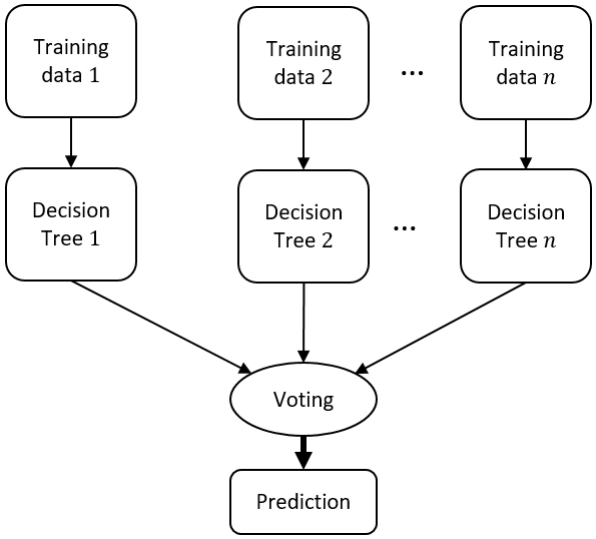

Fig. 4. Random Forest Tree Architecture.

5) Artificial neural network: ANN is composed of three layers: input, output, and hidden layer(s) as shown in Fig. 5 [27]. The input layer nodes communicate with the hidden layer nodes, as do the output layer nodes from each hidden layer node. The network data are taken from the layer of input. The hidden layer receives raw data from the input layer and processes it. The value obtained is transferred to the output layer, which also processes and returns data from the hidden layer. Incapable of justifying its choices is ANN's most critical shortcoming. Here, we have used the MutilayerPerceptron function in WEKA Tool.

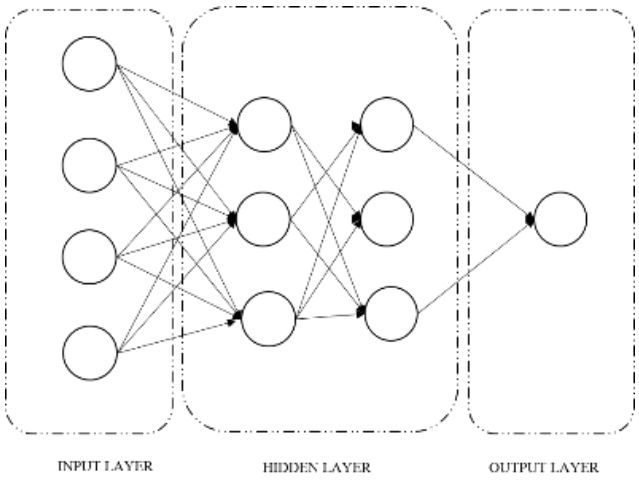

Fig. 5. Simple ANN Architecture.

6) Logistic regression: LR uses sigmoid function instead of linear function as shown in Fig. 6 [28]. In Fig. 6, y represents linear regression and probability p represents LR. The vertical axis is the likelihood of a particular number, and the horizontal axis represents the value of x. A sigmoid function is used by the logistic function to limit the y value from a wide-scale to inside the range (0, 1). Here, we have used the SimpleLogistics function in WEKA Tool.

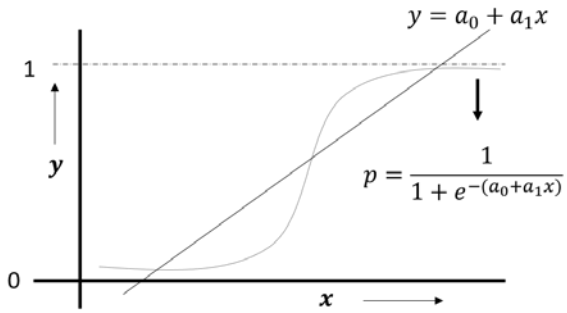

Fig. 6. Graphical Comparison of LR and Linear Regression.

7) Adaptive boosting: Adaptive Boosting (AdaBoost) is an ensemble learning technique that is used to enhance the accuracy of weak binary classifiers i.e., DT. Unlike RF, here weak classifiers add sequentially. For Dataset having number N feature variables, N decision stumps will create. Initially, all decision stumps assigned equal-weighted data. The selection of the base model (first stump) will be based on the lesser value of Entropy. After that, each observation updates with normalized new weight based on performance and total error. Finally, based on a random number and normalized weight a new decision stump will select, and so on. In WEKA Tool, Implementation of Adaptive Boosting is known by AdaBoostM1.

8) k-Nearest Neighbors: k-NN is a classifier that classifies data points based on their closest neighbours. Implementation of k-NN consists of simple steps. Initially, data points transform into vectors. In the next step, the distance between vector points is found by using a mathematical equation such as Euclidian Equation, and Manhattan distance shown in Equation (2). Then the probability of these points calculates being like test data. Finally, the classification of these vector points having the highest probability. Here we have used the IBk (Instance-Based Learner) (Implementation of k-NN) function in WEKA Tool.

$$d(p,q) = \sum_{i=1}^{t} |p_i - q_i| \quad (2)$$

where $d(p,q)$ is the distance between vector p and q; t denotes the number of data points in the vector.

C. Software used

WEKA (Waikato Environment for Knowledge Analysis) is a free and open-source software application designed to address a range of data mining issues [29]. The framework allows the implementation of several algorithms for data analysis and provides an API to call inbuilt algorithms from a particular application by JAVA Programming Language. It provides a variety of tools for classification, regression, clustering, removes irrelevant features, builds associate rules, and visualization of the dataset. We have used WEKA v3.8.5 on Intel® Core™ i3 @ 1.70GHz with 8GB RAM on x64 bit Windows 10 Operating System.

D. Performance Metrics used

1) Confusion matrix: Confusion Matrix represented by $N \times N$ table shown in Fig. 7 that describes how well a classifier performs for which the true values are known. It consists of 4 entities. True positive (TP) are the cases where the classifier predicted that patients have the illness and, they have the illness. True negatives (TN) are those where classifier predicted patient does not have the illness and, they have no illness. False-positive (FP) is also referred to as Type I Error. In this, the classifier predicted that patients have the illness but, they do not have. False-negative (FN) is also referred to as Type II Error. In this case, the Classifier anticipated that the patient would not have the disease, but they do.

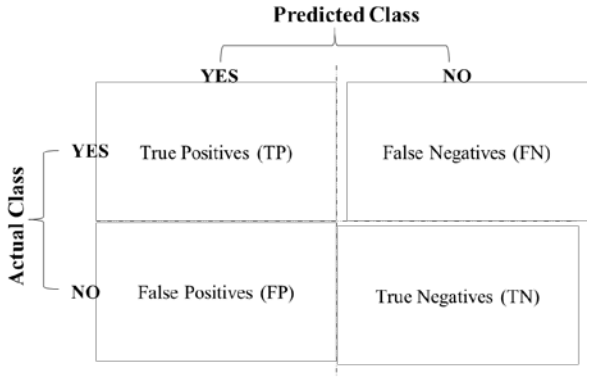

Fig. 7. Representation of Confusion Matrix.

The confusion matrix will then be used to determine Accuracy, Precision, Recall (Sensitivity), and F-Measure. Accuracy means how often is the model correct? Mathematically, it is shown in Equation (3). Precision is defined as the ratio of True Positives to Total Positives and the recall is how many true positives were found by the model. Mathematically, Precision and Recall are shown in Equation (4), Equation (5), respectively.

F-Measure is defined as the Harmonic Mean of Precision and Recall as stated in Equation (6). Instead of balancing the trade-off between Precision and Recall, the researchers can look for a good score of F-Measure. The Receiver Operator Characteristic (ROC) curve is a probability curve that compares the True Positive Rate (TPR) to the False Positive Rate (FPR) at different threshold levels. The greater the ROC Area, the better is the model's ability to differentiate between positive and negative groups.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (3)$$

$$Precision = \frac{(TP)}{(TP+FP)} \quad (4)$$

$$Recall = \frac{(TP)}{(TP+FN)} \quad (5)$$

$$F\ Measure = 2 \times \frac{precision \times recall}{precision + recall} \quad (6)$$

2) Cohen's kappa: These metrics use to measure how closely the instances are classified by the classifier when matched with labelled data as ground truth. It is mathematically shown in Equation (7). The greater the value of Cohen's kappa, the greater will be the level of agreement and the higher will be the percentage of reliable data. A value below 0.60 usually considers a weak classifier.

$$Cohen's\ kappa = \frac{P_a - Pe}{100 - P_e} \quad (7)$$

where P_o is actual percentage agreement, P_e is expected percentage agreement based only on chance.

IV. Experimental Results

This paper examined two research questions for effective and unbiased analyzing the algorithms. To answer RQ1, we have inspected the extensive state-of-the-art related to Predictive algorithms and CVD. Table I clearly showed that *SVM, NB, DT, RF, LR, ANN, AdaBoost and k-NN* are the most common and popular choices for CVD prediction. To answer RQ2, we stated methodology for opting which algorithm outperforms on two similar structured datasets (i.e., Cleveland Dataset and Statlog Dataset). Unlike the Statlog dataset, Cleveland Dataset poses missing values. To remove these missing values, we have applied ReplaceMissingValues Filter in WEKA that replaced these values with modes/means. Later balancing of Datasets has performed by ClassBalancer filter so that each class has the same total weight.

Following data pre-processing, each dataset was divided into Training and Testing data (for validation) using 10-fold cross-validation. Algorithms from RQ1 were applied to these datasets. To measure the effectiveness of these algorithms,

each one was put to the test using performance measures, the results of which were displayed in Table III and Table IV.

Against Cleveland Dataset, the result of the performance analysis showed that both SVM and ANN perform better than other selected algorithms with the accuracy of ~84.15% and ~84.09% respectively (Table III). DT scored 73.62%, Naïve Bayes scored ~81.67%, RF scored ~81.37%, Logistic Regression scored ~81.37%, AdaBoost scored ~82.99%, and k-NN scored ~75.74% in terms of accuracy. The accuracy of the ANN classifier is very close to SVM but the ROC Area value of ANN (0.907) is more than SVM (0.842) (see Table III). So, both ANN and SVM are suitable choices for the prediction of CVD against the Cleveland Dataset.

Analysis Result against Statlog Dataset showed there were three algorithms whose performance was worthy to talk about (Table IV). SVM scored the highest accuracy of ~84.07%. Next in order, NB and LR showed the same accuracy of ~83.70%. DT scored ~76.66%, RF scored ~76.29%, ANN scored ~78.14%, AdaBoost scored 80% and k-NN scored ~75.18% in terms of accuracy. If we compare the ROC area then both NB and LR are better than SVM (see Table IV).

The results discussed were about individual datasets. If we compared the accuracy of algorithms against the Cleveland dataset and Statlog dataset then SVM performed better than other algorithms (see Fig. 8). Against Cleveland Dataset, it showed an accuracy of ~84.15% and Against Statlog Dataset, it showed an accuracy of ~84.07%. Next in order, NB showed an accuracy of ~81.67% against the Cleveland Dataset and an accuracy of ~83.70% against the Statlog Dataset.

TABLE III. PERFORMANCE METRICS OF THE ALGORITHMS AGAINST CLEVELAND DATASET

Algorithms	Accuracy (in %)	Precision	Recall	F1	ROC Area	Kappa
SVM	**84.1568**	0.843	0.842	0.841	**0.842**	0.6831
NB	81.6733	0.817	0.817	0.817	0.899	0.6335
DT	73.6232	0.736	0.736	0.736	0.741	0.4725
RF	81.3702	0.814	0.814	0.814	0.900	0.6274
LR	81.3702	0.814	0.814	0.814	0.900	0.6274
ANN	**84.0909**	0.841	0.841	0.841	**0.907**	0.6818
AdaBoost	82.9974	0.830	0.830	0.830	0.892	0.6599
k-NN	75.7444	0.758	0.757	0.757	0.750	0.5149

TABLE IV. PERFORMANCE METRICS OF THE ALGORITHMS AGAINST STATLOG DATASET

Algorithms	Accuracy (in %)	Precision	Recall	F1	ROC Area	Kappa
SVM	**84.0741**	0.841	0.841	0.840	**0.785**	0.6762
NB	**83.7037**	0.837	0.837	0.837	**0.898**	0.6683
DT	76.6667	0.766	0.767	0.767	0.744	0.5271
RF	76.2963	0.764	0.763	0.763	0.762	0.5216
LR	**83.7037**	0.837	0.837	0.837	**0.900**	0.6683
ANN	78.1481	0.784	0.781	0.782	0.839	0.5601
AdaBoost	80.0000	0.800	0.800	0.800	0.878	0.595
k-NN	75.1852	0.753	0.752	0.752	0.750	0.4988

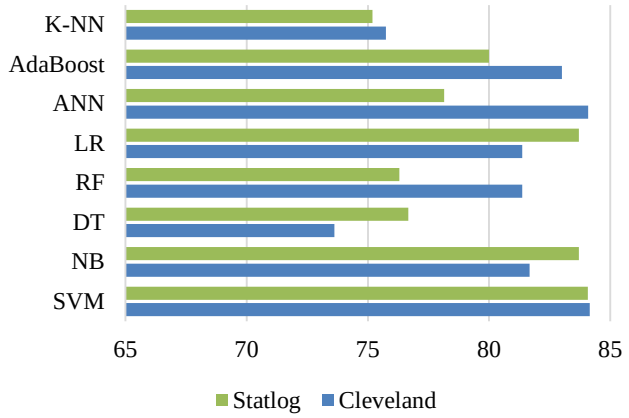

Fig. 8. Comparison of Accuracy (in %) against Cleveland and Statlog Dataset.

Algorithms having a ROC Area value near 1 generally consider a good classifier against the dataset. LR scored a ROC Area of 0.9 against both datasets (see Fig. 9). Next in order, ANN showed 0.907 against Cleveland Dataset and 0.839 against the Statlog Dataset.

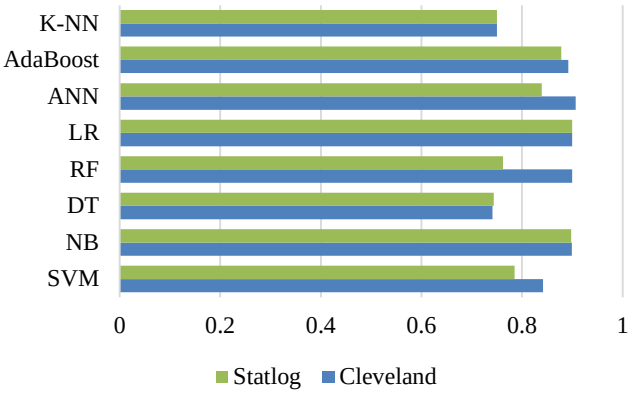

Fig. 9. Comparison of ROC Area against Cleveland and Statlog Dataset

V. CONCLUSION AND FUTURE WORK

Predictive Algorithms founds to be very effective in the automatic prediction of CVD. In this work, we analyzed popular predictive algorithms namely SVM, NB, DT, RF, LR, ANN, AdaBoost and k-NN. They were chosen based on the state-of-the-art related to the CVD and Predictive Algorithms. The experiment was conducted using two similar structured datasets (i.e., Cleveland and Statlog Dataset) on open-source WEKA software. The outcome of the experiment concluded that (1) SVM showed maximum accuracy against the datasets, (2) LR showed a ROC Area of 0.9 against both the datasets. These results imply that (1) SVM shows better accuracy against most of the datasets by finding optimal hyperplane using kernel tricks, (2) LR shows better ROC Area against the binary classification datasets.

These findings will help the researchers and Health institutions (1) To understand the current trends related to CVD prediction using the algorithm(s), (2) To build successful and

effective CDPS (i.e., Clinical Disease Prediction System) for CVD. Unfortunately, we were unable to study and analyze hybrid models/algorithms but it can extend in future by considering this work as a blueprint/base. Future work should give priority to (1) Real-time and Complexed CVD data, (2) Ensemble Learning and Hybrid Models for analysis, (3) Checking the effects on the value of Performance Metrics against different validation and features selection techniques.

REFERENCES

[1] "India | Institute for Health Metrics and Evaluation." http://www.healthdata.org/india (accessed Mar. 20, 2021).

[2] "Cardiovascular diseases (CVDs)." https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (accessed Aug. 15, 2020).

[3] C. Abbafati et al., "Global burden of 369 diseases and injuries in 204 countries and territories, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019," Lancet, vol. 396, no. 10258, pp. 1204–1222, Oct. 2020, doi: 10.1016/S0140-6736(20)30925-9.

[4] Aman and R. S. Chhillar, "Disease predictive models for healthcare by using data mining techniques: State of the art," SSRG Int. J. Eng. Trends Technol., vol. 68, no. 10, pp. 52–57, 2020, doi: 10.14445/22315381/IJETT-V68I10P209.

[5] A. Muniasamy, V. Muniasamy, and R. Bhatnagar, "Predictive analytics for cardiovascular disease diagnosis using machine learning techniques," in Advances in Intelligent Systems and Computing, Feb. 2021, vol. 1141, pp. 493–502, doi: 10.1007/978-981-15-3383-9_45.

[6] J. Deshmukh, M. Jangid, S. Gupte, and S. Ghosh, "Heart disorder prognosis employing knn, ann, id3 and svm," in Advances in Intelligent Systems and Computing, Feb. 2021, vol. 1141, pp. 513–523, doi: 10.1007/978-981-15-3383-9_47.

[7] S. B. Garg, P. Rani, and J. Garg, "Performance analysis of classification methods in the diagnosis of heart disease," in Lecture Notes in Networks and Systems, 2021, vol. 140, pp. 717–728, doi: 10.1007/978-981-15-7130-5_58.

[8] R. Katarya and S. K. Meena, "Machine Learning Techniques for Heart Disease Prediction: A Comparative Study and Analysis," Health Technol. (Berl)., vol. 11, no. 1, pp. 87–97, Jan. 2021, doi: 10.1007/s12553-020-00505-7.

[9] I. Karun, "Comparative Analysis of Prediction Algorithms for Heart Diseases," in Advances in Intelligent Systems and Computing, 2021, vol. 1158, pp. 583–591, doi: 10.1007/978-981-15-4409-5_53.

[10] 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, pp. 107562–107582, 2020, doi: 10.1109/ACCESS.2020.3001149.

[11] A. Singh and R. Kumar, "Heart Disease Prediction Using Machine Learning Algorithms," Int. Conf. Electr. Electron. Eng. ICE3 2020, pp. 452–457, 2020, doi: 10.1109/ICE348803.2020.9122958.

[12] G. Choudhary and S. Narayan Singh, "Prediction of heart disease using machine learning algorithms," in Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics, ICSTCEE 2020, Oct. 2020, pp. 197–202, doi: 10.1109/ICSTCEE49637.2020.9276802.

[13] P. S. Sangle, R. M. Goudar, and A. N. Bhute, "Methodologies and Techniques for Heart Disease Classification and Prediction," Jul. 2020, doi: 10.1109/ICCCNT49239.2020.9225673.

[14] D. Shah, S. Patel, and S. K. Bharti, "Heart Disease Prediction using Machine Learning Techniques," SN Comput. Sci., vol. 1, no. 6, p. 345, Nov. 2020, doi: 10.1007/s42979-020-00365-y.

[15] C. C. Peng, C. W. Huang, and Y. C. Lai, "Heart Disease Prediction Using Artificial Neural Networks: A Survey," in 2nd IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability 2020, ECBIOS 2020, May 2020, pp. 147–150, doi: 10.1109/ECBIOS50299.2020.9203604.

[16] H. El Hamdaoui, S. Boujraf, N. E. H. Chaoui, and M. Maaroufi, "A Clinical support system for Prediction of Heart Disease using Machine Learning Techniques," Sep. 2020, doi: 10.1109/ATSIP49331.2020.9231760.

[17] N. K. Kumar, G. S. Sindhu, D. K. Prashanthi, and A. S. Sulthana, "Analysis and Prediction of Cardio Vascular Disease using Machine Learning Classifiers," in 2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020, Mar. 2020, pp. 15–21, doi: 10.1109/ICACCS48705.2020.9074183.

[18] J. Santhana Krishnan and S. Geetha, "Prediction of Heart Disease Using Machine Learning Algorithms.," Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. ICIICT 2019, pp. 1–5, 2019, doi: 10.1109/ICIICT1.2019.8741465.

[19] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," IEEE Access, vol. 7, pp. 81542–81554, 2019, doi: 10.1109/ACCESS.2019.2923707.

[20] A. N. Repaka, S. D. Ravikanti, and R. G. Franklin, "Design and implementing heart disease prediction using naives Bayesian," Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019, vol. 2019-April, no. Icoei, pp. 292–297, 2019, doi: 10.1109/icoei.2019.8862604.

[21] "UCI Machine Learning Repository: Heart Disease Data Set." https://archive.ics.uci.edu/ml/datasets/Heart+Disease (accessed Aug. 15, 2020).

[22] "UCI Machine Learning Repository: Statlog (Heart) Data Set." http://archive.ics.uci.edu/ml/datasets/statlog+(heart) (accessed Jul. 12, 2021).

[23] C. Cortes and V. Vapnik, "Support-Vector Networks," Mach. Learn., 1995, doi: 10.1023/A:1022627411411.

[24] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," Mach. Learn., 1997, doi: 10.1023/a:1007465528199.

[25] J. R. Quinlan, "Induction of decision trees," Mach. Learn., 1986, doi: 10.1007/bf00116251.

[26] L. Breiman, "Random forests," Mach. Learn., 2001, doi: 10.1023/A:1010933404324.

[27] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," Heliyon. 2018, doi: 10.1016/j.heliyon.2018.e00938.

[28] L. J. Davis and K. P. Offord, "Logistic regression," in Emerging Issues and Methods in Personality Assessment, 2013.

[29] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." https://www.cs.waikato.ac.nz/ml/weka/ (accessed Aug. 15, 2020).