Impact of Preprocessing Methods on Healthcare Predictions

Puneet Misra[1] and Arun Singh Yadav[2]

Abstract—Machine learning (ML) is now a day gaining immense importance and is becoming a key technology as the rapid growth of quality of medical data and information. But the early and accurate detection of disease is still a challenge due to the complex, incomplete and multidimensional healthcare data. Data preprocessing is an essential step of ML whose primary goal is to provide processed data to improve the prediction accuracy. This study summarizes the popular data preprocessing steps based on their usages, popularity and literature. After that the selected preprocessing methods is applied on the raw data which is then used by classifiers for predictions. In this experiment we have taken diabetes classification problem. Type II diabetes mellitus (T2DM) is a major disease with high penetrance in human around the world and still rising. This may cause other serious complications like kidney failure, heart failure, blindness etc. The early detection and diagnosis help to identify and may avoid these complications. Several classification algorithms exist but selecting the best classifier surely improves the accuracy of the predictions. The preprocessing methods selected in this study are Multiple Imputation, k-means for missing values treatment, Discretization to change in discrete values, Standard scaler, Min-Max scalar for feature scaling and, Random Forest (RF) for feature selection. For the classification Logistic Regression (LR), Artificial Neural Network (ANN), Support Vector Machine (SVM) and Random Forest (RF) are used. To evaluate the performance of model accuracy, sensitivity, specificity are used. This study compares the model performance with and without preprocessed data and has proved that the selected preprocessing methods significantly improves the model performance.

Keywords: Machine learning, Disease prediction, Classification, Preprocessing, Multiple Imputation, k-NN, Standard Scaler, Min-Max scalar, RFE, LR, ANN, SVM, RF

I. INTRODUCTION

Machine Learning (ML) is gaining importance day by day due to its capability work with heterogeneous data set. ML algorithms directly learn from the data, produce the hidden insights and can predict or forecast the future outcomes on the basis of its learning[1]. Predictions can be done either by classification or regression approach. The classification accuracy of the predictions depends on the quality of the data. Data generated from various sources may have missing values, noisy, inconsistent, voluminous and class imbalanced[2]. This imperfect data requires data preparation stage to clean and prepare the data[3] for further analysis. To get quality of data, machine learning provides one of the most meaningful steps called data preprocessing. This step usually takes the significant amount of time[4] and should be implemented carefully to improve the overall model performance.

The data preprocessing task includes certain steps like data preparation, integration, cleaning, normalization, scaling and data reduction techniques to reduce complexity, to noisy and irrelevant elements using feature selection and discretization etc. After this the outcome expected a final dataset for further analysis using ML algorithms.

[1,2]*Department of Computer Science, University of Lucknow, Lucknow, India-226007*
E-mail: [1]puneetmisra@gmail.com, [2]arun.ai.lkouniv@gmail.com

The main objective of this paper to study the impact of selected preprocessing techniques on prediction value and justify that data preprocessing using intelligent techniques significantly improves the model performance. Another objective is to study various preprocessing methods and select the best among them and this selection has been done on the basis of their usages, popularity and should have been widely cited by research community.

We have started with the study of the most influential data preprocessing algorithms under various section of it and select the widely used among them. In next section we discussed about the dataset one public and another real-world. The experimental setup and proposed framework is explained in next section. We further described the model evaluation. The detailed discussion on the result and performance of the model before and after applying data processing techniques is covered in next section and the last section covered the conclusion and future work.

II. STUDY OF DATA PROCESSING ALGORITHMS

A. Data Cleaning

The data taken from the real-world problem is seldom clean and complete, specially the healthcare field. The data cleansing/cleaning step deals with the treatment of missing values, errors and inconsistencies in the dataset. To deal with this issue various methods or techniques are evolved since past. Choosing the appropriate technique depends on various factors[5]. Proper handling of imperfect data involved certain steps described below:

1) Missing Value Treatment

Missing value (MV) is defined as the data value that is not present in the cell of the corresponding column. The reasons of MVs in healthcare context can be the human omission, not applicable instance, not recorded electronically by the sensor, patient not present on the ventilator due to a medical decision, the patient condition is unrelated for a particular variable, electricity failure, database synchronization and others. Any statistical analysis or machine learning activity on the data having MVs may produce the undesired or biased result and improper handling of missing data can produce misleading conclusions[6]. According to the survey[6] problems associated with the MVs can be loss of efficiency, complications in handling and analyzing data and biased result. But the incomplete data is the subject to study due to its impact on the classification accuracy[7]. Before applying any treatment first, we need to understand the pattern of missingness. Researchers Little and Rubin categorize it into three ways[8]: Missing completely an random(MCAR), Missing at random(MAR) and Not Missing at random(NMAR). The first one occurs when the missing data does not depend on any other attribute of the dataset, second occurs when the distribution of MVs of an attribute depends on observed data of another attribute but itself and the third one happens when the distribution of MVs of an attribute depends on itself. Handling of MVs can be done either discarding it or imputing the missing data.

2) Discarding the Missing Values

The most usual approach to discard the MVs but this approach is not so practical[9] because if the train data have a large number of missing values then the produced result must biased. If the dataset has a small number of missing values, then we must assure that analysis on the remaining part will not produce the inference bias. The deletion of the MVs can be done in following ways:

- *Listwise deletion (Deleting rows):* In this case the complete case analysis is done and removes all the observed cases having more than one missing value. But this approach is helpful in small number of missing cases. It works well when dataset has the MCAR missing pattern and which is rare.
- *Pairwise deletion:* This method attempts to minimize the error in list wise deletion. In this the attribute with MVs is deleted if it is not use as a case of another attribute while analyzing the data. It strengthens the analysis power but create other complications like producing standard error.
- Dropping attribute completely: This is very rare and, in my opinion, sometimes you can drop the attribute completely if the missing observations are more than 60% and the attribute looks insignificant in the analysis. Sometime attributes with missing values should be kept due to high relevance.

Dropping the missing cases or attribute will never be the best choice because it may contain some meaningful insights in data analysis. Hence some probabilistic method and imputation methods always get priority in MVs treatments.

3) Parameter Estimation Method

Traditionally statisticians used the probabilistic approach to estimate the MVs. In this the maximum likelihood procedure is used to estimate the parameters of the model for complete dataset. To optimize the maximum-likelihood procedure R.A. Fisher introduce a meta algorithm Expectation-Maximization[8]. It estimates the parameters of a probability distribution which better fit the observed data in the iterative processes[10]. But its convergence rate is slow and faces issue to deal with non-linear high-dimensional data. This method also underestimates the standard errors at the time of estimation process.

Multiple Imputation (MI): Rubin (1976) first developed a framework for incomplete data after some years in 1987 he proposed a MI method. The exhaustive discussion on Multiple Imputation methods can be found in Schafer et al. research article[9] . It has a clear advantage over the parameter estimation method and single imputation method. Here the missing terms is imputed more than once from the Bayesian posterior distribution, which maintains the variation in the dataset. It additionally computes the variation-based error, sometimes called 'between imputation error'. MI is very simple and can be implemented in so many ways like MCMC (Markov Chain Monte Carlo) and MICE (Multiple Imputation by Chained Equations) methods are widely used. The MCMC is a simulation-based method while MICE is multivariate imputation method and gives better result on MAR data. MICE use series of regression models to compute the missing data where type of regression is depending on the type of data. If the missing variable distribution is continuous then linear regression and if its distribution is in binary form logistic regression is used[14].

4) K–Nearest Neighbor Imputation (KNNI)

The traditional statistical models are working fine for MVs imputation with its limitations as the nature of data, percentage of missing data etc. In the same series authors G. Batista and M. Monard proposed machine learning technique as an imputation method. Authors uses k-NN algorithm as an imputation method with three other imputation techniques mean, mode substitution and C4.5, CN2[11] on the three different healthcare problem. All the four methods were analyzed with different percentage of missing data of three datasets. This research states that all the imputation techniques work well but k-NN algorithm as 10-NNI outperform on the breast cancer problem. k-NN[15] is most widely used clustering algorithm[16] where k-neighbors are chosen on the basis of distance measure(Euclidean, Hamming, Manhattan, etc.). It selects

the similar features values and creates the distance matrix and on the basis of that it creates the clusters of similar values. It can work on both discrete (Hamming distance) and continuous data (Euclidean distance). There is no specific criterion to select the best k-value, but elbow rule helps here to decide best k-value. From the machine learning point of view k-NN as an imputation is an intuitive solution[17]. Unlike the other discussed imputation method this algorithm doesn't need a model to predict the MVs hence it is more popular and widely used.

B. Feature Scaling

Feature scaling have significant impact on some algorithms[18] (PCA, KNN, SVM and NN, if use gradient descent optimization) while minimal on others. The features of the dataset may vary in terms of magnitude, range and unit. Most of ML algorithms work on the magnitude of the measurement not on the unit. ML algorithms uses Euclidean distance between data points so the distance measure of a higher magnitude feature and low magnitude feature would produce undesired results. Hence it is the necessity to scale all the features at same level. Data scaling can be done either by Z-score algorithm or Min-Max algorithm. The z-score algorithm (standardization) scaled the features centered near to zero i.e. mean of the distribution μ=0 and standard deviation σ=1. Here the original value is replaced by $\dot{x} = \frac{x-\bar{x}}{\sigma}$. Min-Max scaling is another approach which scaled the feature to a fixed range either [-1, 1] or [0, 1]. This takes minimum and maximum sample as [x_{max} & x_{min}] and replaced the original features to scaled features as $\dot{x} = \frac{x-x_{min}}{x_{max}-x_{min}}$.

C. Data Reduction

Data is generating in a rapid pace through various electronic devices which is very large and high dimensional in nature, especially medical data (patient information, lab test, symptoms, device generated etc.)[10]. Handling multidimensional is a tedious task and prediction with all the features sometimes may not be useful because the presence of irrelevant features. Hence selecting the best features or generating the new features from existing can improve the classifier accuracy. This introduced the data reduction task to select most relevant predictors either by feature selection(FS), extraction(FE) and discretization methods[10]. [19] Study shows that more than 75% of data reduction task is concerned feature selection, about 15% to feature extraction and less than 10% concerned discretization method. Discretization converts the quantitative data into qualitative data with certain number of intervals. Discretization is getting importance in research during recent years[20]. It becomes the necessity where algorithm works well on nominal data like Decision tree[21], Naïve Bayes[22]. Discretization simplifies the data and makes learning faster and accurate where one more advantage is discrete variables are easy to understand and interpret and also reduces model complexity. One of the major drawbacks with this technique is the loss of information. FE is another data reduction technique which generates the new and stronger features that has the strong impact on the classifier result. Principal Component Analysis(PCA)[23] is one of the most widely used linear method in this category in context of ML where Factor Analysis(FA)[24] is second on the number and widely used in statistical learning. Instead of dropping weak predictors PCA generate new predictors uncorrelated to each other. But generally PCA outperform if dataset contains independent but uncorrelated predictors and one more issue with this is the selection of no. of principal components[25]. Feature selection(FS) technique does not make any change in the original feature which helps in the interpretation and understanding[26]. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible[27]. The detail discussion on FS technique, its importance, benefits can be seen in this article[28]. It is one of the most researched step under data preprocessing since past[29][30]. Feature subset selection can be done either by filter or wrapper approach. Wrapper method use ML to decide the best feature by train it while filter method don't use the ML. Filter method is fast, but it is not the best choice in case of small dataset while wrapper method is slow, computationally costly but works well massive or small dataset. In this study Random Forest (RF) method is used to select the best subset of features[31]. RF ranks the features using ensemble of decision tree and other tree-based approaches. Each node of the decision tree signifies a condition on single feature, and it divides it into two parts, so the similar response values remains in the same set. It selects the optimal feature on measure of impurity. In each training step it checks the how much each feature decreases the weighted impurity of decision tree.

III. DATASET DESCRIPTION

In this study we have taken two different diabetes type-2 dataset as a classification problem and applied selected preprocessing techniques to check the impact of data preprocessing on the classifier predictions accuracy. The first dataset PIMA Indian Diabetes Dataset (PIDDs)[32] is taken from the Machine Learning Database repository of University of California, Irvine (UCI)[33] and the second one LUDB2 from the Bioinformatics Research Lab, Lucknow University. Some previous studies show that Pima Indians may be genetically predisposed to type-2 diabetes which was 19 times of the any nearby typical town. The PIDDs (table1) contain total 9 features where 8 features are used as predictor variables and last one class variable indicating the onset of diabetes within 5 years. The dataset contains total 768 female patients of age 21 years and above where 268 instances are diabetic and 500 instances are non-diabetic.

The LUDB2 dataset (table 2) contain 12 features where 11 features are used as independent variables, means the predictors and last feature is the dependent variable or class variable which indicates the diabetes status (yes/ no). The dataset contains 1000 instances of patient's data where 500 patients are non-diabetic and rest 500 are diabetic. In this dataset family history is taken as a one categorical variable to represent the heredity in the family.

Table 1: Pidds type 2 diabetes dataset

S.No.	Columns	Variables	Description
1	AGE	Age (years)	Age of the female patients [21 to 81]
1.	PREG	Pregnancy	No. of times a woman gets pregnant [0 to 17]
2.	BMI	Body mass index	Body Mass Index (Weight in kg/height in m^2) [0 to 67.1]
3.	PGLU	Plasma Glucose concentration	Plasma Glucose concentration measured using a 2-hour oral glucose tolerance test [0 to 199]
4.	DBP	Diastolic blood pressure	Diastolic blood pressure (mm Hg) [0 to 122]
5.	INSU	Insulin	2-hour serum insulin (mu U/ml) [0 to 846]
6.	TFST	Triceps skin fold thickness	Triceps skin fold thickness(mm) [0 to 99]
7.	DPF	Diabetes pedigree function	Diabetes pedigree function [0.08 to 2.42]
8.	CLASS	Diabetes status (0 or 1)	Patient with diabetes onset within five years

Table 2: LUDB2 dataset Type-2 Diabetes Dataset with variable description

S.No.	Columns	Variables	Description
1	GNDR	Gender	Male and Female
2	FHIST	Family History	anyone in family have db2 positive or not
3	AGE	Age (Years)	20 to 70 years
4	BMI	BMI (kg/m2)	Body mass index
5	WHRATIO	W/H ratio	Waist-Height Ratio
6	SBP	SBP (mmhg)	Systolic Blood Pressure
7	DBP	DBP (mmhg)	diastolic blood pressure
8	FPG	FPG (mg/dl)	Fasting Plasma Glucose
9	PPG	PPG (mg/dl)	Postprandial Blood Glucose
10	HBA1C	Glucose HbA1c (%)	hmoglobin A1C (HbA1c)
12	CLASS	Diabetic Status (Months)	Positive or Negative

IV. Experimental Setup

The whole experiment is divided into two parts: prediction of classifier accuracy with and without preprocessing techniques. In first part we have used the dataset directly to train and test the Neural Network classifier and store the result in table 3. As previous studies have shown that the prediction accuracy depends on the quality of data. The quality of data can be check either exploring the datasets graphically or analytically. So, in second part we have done exploratory data analysis to identify the MVs, noisiness in the given datasets and then applied imputation technique *k-NN* for MVs treatment and store it in two different variables. After dealing with MVs the scaling of features is done by *MinMax* Scalar because features vary in magnitude and nature (categorical and non-categorical). Selection of best features is the next hurdle, so we have used RF as a feature selection. This preprocessed data then applied to different classifier to predict the diabetic and non-diabetic patient, see table V & VI for the results. All the experimental work has been done on *Jupyter Notebook v. IDE*. The *Python 3.0* programming language is used to for the analysis and for building prediction models. We have used *numpy, pandas, matplotlib* libraries of python 3.0 for dataset representation, processing and visualization and *scikit-learn* for building the machine learning models for predictions.

V. Evaluation Matrix

To evaluate the model performance Accuracy, sensitivity, specificity are used. Accuracy measure is always preferred choice in classification problem if target variable in dataset is approximately balanced. In our case LUDB2 dataset have 50% diabetic and rest 50% are non-diabetic but PIDDs is imbalanced and contains 70% positive and 30% negative cases. To calculate performance measures Accuracy, sensitivity and specificity certain matrix is needed i.e. True Positive (TF), True Negative (TN), False Positive (FP) and False Negative (FN). The formulation of these matrices is given in table below:

Table 3: Model evaluation metrices

Matric	Formula
Accuracy	(TP+TN)/(TP+TN+FP+FN)
Sensitivity	TP/(FN+TP)
Specificity	FP/(FP+TN)

VI. Result and Discussion

The PIDDs dataset, when used without preprocessing as an input to the classifiers LR (Logistic Regression), ANN (Artificial Neural Network), SVM (support vector machine) and Random Forest, models exhibit the accuracies 75%, 67%, 65% and 74%. Here, it can be observed that Logistic regression model gives the best response without applying any preprocessing methods while Random forest is on second top performer. But ANN and SVM is not performing well. The reason why they are giving the worst result can be missing values and noise and both models requires scaled values. But when we applied same methodology on LUDB2 dataset, all the classifiers exhibit good accuracies 98%, 94%, 74% and 100%. So here Random Forest is outperforming and can

classify positive and negative cases with 100% accuracy. The reason of this can be LUDB2 dataset have no missing values, no noise and balanced. But after applying the preprocessing methods on both the dataset and when applied on classifiers, results got improved. We start the experiment by investigating the missing values present in the dataset and it is found that LUDB2 dataset have no missing data values. But FIHST attribute have some missing data as ***NaN*** which means that no one is diabetic in his family, hence we simply replace it with value ***zero (0)***. But when we explored PIDDs dataset graphically and analytically, it contains 4% to 48% missing or incorrect values (refer to table IV).

Table 4: Incorrect or missing value instances present in different attributes of PIDDs dataset out of 786 rows

S.No.	Attribute	No. of Missing or incorrect values
1.	PREG	5
2.	BMI	35
3.	PGLU	227
4.	DBP	374
5.	INSU	11

The above table shows that DBP (Diastolic Blood Pressure) contains 48% incorrect or missing values (only those instances that have value 0) and PGLU (Plasma Glucose concentration) have 29% missing instances. Theoretically 25% to 30% missing values are accepted, but we cannot delete them because we don't know how much they impacts the predictions. We replaced missing instances with *kNN* method but in both the datasets have extreme values that may impact the learning process of the model, but the treatment of these outliers is out of the scope of this study, so we as it is keep them in dataset. In LUDB2 dataset *GNDR & FHIST* attributes are of object type so we convert them into integer type but when we applied this on simple logistic regression model, we found that variability is not impacting the accuracy, so we discretize it into 0 & 1. The same case is done with PIDDs dataset the attribute GNDR to 0 & 1. The same case we have found with *PREG* attribute in PIDDs dataset, so discretize it into 0 & 1.

We have used distance-based classification algorithms that's why we need to scale the feature for better accuracy. So, for this we have used Min-Max Scaler and Standard Scaler method in both the datasets PIDDs and LUDB2.

For feature selection first we checked the feature importance and after that we selected the best among them. After applying Random Forest algorithm on PIDDs to calculate feature importance, we have found that features PGLU (Plasma Glucose concentration) is getting immense importance, INSU(Insulin) is on second place, BMI on third and AGE scored fourth place. But BMI is highly correlated with TFST (skin thickness). DBP had the highest missing values (table 3) and has lowest ranking in feature estimators.

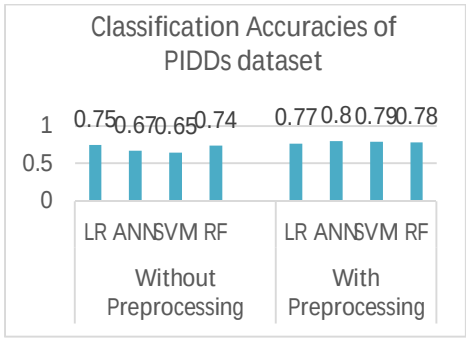

Figure 1: Comparison of classification accuracies before and after applying preprocessing methods in PIDDs dataset

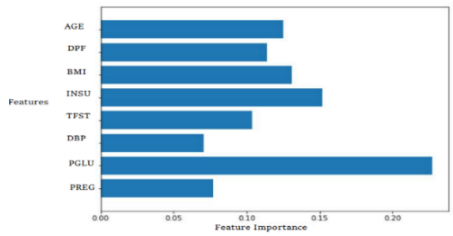

Figure 2: Feature importance using random forest in PIDDs dataset

DPF (Diabetes Pedigree function) is getting good importance but showing very less correlation with class attribute. Hence, we have not taken these three (DBP, DPF and TFST) attributes for further analysis. So, we have done the next analysis by using PGLU, AGE, BMI, INSU and PREG. The same process when we have done with LUDB2 dataset we have found that HbA1C, FPG, PPG, W/H ratio, BMI, SBP and AGE are the top rankers (see fig. 2).

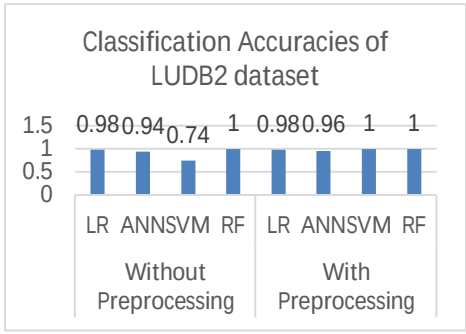

Figure 3: Comparison of classification accuracies before and after applying preprocessing methods in LUDB2 dataset

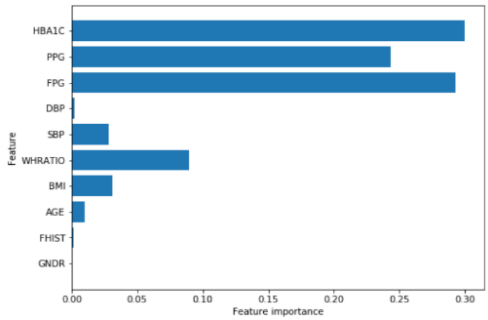

Figure 4: Ranking of features using random Forest in LUDB2 dataset

After applying all the preprocessing methods on both the datasets a significant amount of increase can be seen by all the classifiers in the accuracies in PIDDs dataset (refer to table V). While LUDB2 dataset is showing the small increase in ANN 94% to 96% but LR and Random Forest classifiers has no change in accuracy, but sensitivity and specificity interchangeably changes in LR. But SVM achieved the highest change in accuracy from 74% (without preprocessing) to 100% (with preprocessing) (refer to table VI).

The ANN models exhibit the highest change in accuracy in PIDDs dataset from 67% (without preprocessing) to 80% (with preprocessing) where SVM is second on the list and achieved 79% of accuracy. We can draw one more observation from here that LR is the less complex model, but it is also able to classify the diabetic and non-diabetic patients with 77% (PIDDs) and 98% (LUDB2) after applying the preprocessed dataset, which is notable.

The comparative analysis of the accuracies achieved by various classifiers on two different datasets can be seen in the fig. 3 and fig. 4 while the detailed performance analysis can be seen in table V and table VI.

Table 5: Outcome of classifiers before and after applying pre-processing techniques on PIDDs datasets

Preprocessing Techniques	Classifier	Accuracy	Sensitivity	Specificity
No Preprocessing	LR	0.75	0.57	0.86
With Preprocessing	**LR**	**0.77**	**0.56**	**0.88**
No Preprocessing	ANN	0.67	0.25	0.89
With Preprocessing	**ANN**	**0.80**	**0.65**	**0.88**
No Preprocessing	SVM	0.65	0	1
With Preprocessing	**SVM**	**0.79**	**0.82**	**0.78**
No Preprocessing	Random Forest	0.74	0.48	0.89
With Preprocessing	**Random Forest**	**0.78**	**0.67**	**0.84**

Table 6: Outcome of classifiers before and after applying pre-processing techniques on LUDB2 datasets

Preprocessing Techniques	Classifier	Accuracy	Sensitivity	Specificity
No Preprocessing	LR	0.98	0.96	1.0
With Preprocessing	**LR**	**0.98**	**1.0**	**0.96**
No Preprocessing	ANN	0.94	0.88	1.0
With Preprocessing	**ANN**	**0.96**	**0.92**	**1.0**
No Preprocessing	SVM	0.74	1.0	0.48
With Preprocessing	**SVM**	**1.0**	**1.0**	**1.0**
No Preprocessing	Random Forest	1.0	1.0	1.0
With Preprocessing	**Random Forest**	**1.0**	**1.0**	**1.0**

VII. CONCLUSION

This work proposes the impact and importance of selected preprocessing methods on the healthcare predictions. We have used two different datasets of healthcare sector for predicting the type 2 diabetes onset. Firstly, we started with the detailed study of different preprocessing methods, its impacts on the predictions, and its usability with limitations in different scenarios. We have found that traditional missing data imputation methods were inefficient with large set of missing values in a specific attribute while machine learning based kNN methods has worked well in this scenario. One more conclusion has seen that different distance-based classifiers worked differently with Standard Scaler and Min Max Scaler. Study also shows that feature selection is the preferred method in data reduction and tree-based methods are widely used with small dataset. The experimental study on public and private dataset have some interesting conclusions. The preprocessed PIDDs dataset shows an average accuracy of 80% with ANN classifier, as against the accuracy of without preprocessed dataset which is 67%. The improved accuracy is very near 79% with SVM on tested modified dataset. Hence, ANN is the top performer on PIDDs dataset.

When the same experiment has done with real private LUDB2 dataset, which has no missing values, and after preprocessing, all the classifiers are showing the similar accuracies with small increment. Almost all the classifiers achieved more than 90% accuracy expect SVM on original LUDB2 dataset. But in feature selection it is found that only Hb1AC and FPG attributes can classify the patient accurately, but we have taken more for the caution. Hence the preprocessed LUDB2 dataset improved the accuracies a little but SVM shows the drastic increase from 74% to 99%.

This proposed approach is tested on similar disease health data with some common attributes and has shown that selected preprocessing methods surely improves the predictions. Some more studies and experiment can do with other healthcare prediction problems and can be used on any prediction model by applying the same approach.

REFERENCES

[1] S. Ben-David and S. Shalev-Shwartz, Understanding Machine Learning: From Theory to Algorithms. 2014.

[2] S. Batra and S. Sachdeva, “Organizing standardized electronic healthcare records data for mining,” Heal. Policy Technol., 2016.

[3] R. Duggal, S. Shukla, S. Chandra, B. Shukla, and S.K. Khatri, “Impact of selected pre-processing techniques on prediction of risk of early readmission for diabetic patients in India,” Int. J. Diabetes Dev. Ctries., vol. 36, no. 4, pp. 469–476, 2016.

[4] wp185007, Data Preparation for Data Mining.doc. 2006.

[5] F. Cismondi, A.S. Fialho, S.M. Vieira, S.R. Reti, J.M. C. Sousa, and S.N. Finkelstein, “Missing data in medical databases: Impute, delete or classify?,” Artif. Intell. Med., 2013.

[6] H. Wang and S. Wang, “Mining incomplete survey data through classification,” Knowl. Inf. Syst., Vol. 24, No. 2, pp. 221–233, 2010.

[7] I.A. Gheyas and L.S. Smith, “A neural network-based framework for the reconstruction of incomplete data sets,” Neurocomputing, vol. 73, no. 16–18, pp. 3039–3065, 2010.

[8] D.B.R. Roderick J.A. Little, “Statistical Analysis with Missing Data,” WILEY Ser. Probab. Stat., p. 96, 2010.

[9] J.L. Schafer and J.W. Graham, "Missing data: Our view of the state of the art," Psychol. Methods, vol. 7, no. 2, pp. 147–177, 2002.

[10] S. García, J. Luengo, and F. Herrera, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining," Knowledge-Based Syst., Vol. 98, pp. 1–29, 2016.

[11] G.E.A.P.A. Batista and M.C. Monard, "An analysis of four missing data treatment methods for supervised learning," Appl. Artif. Intell., Vol. 17, No. 5–6, pp. 519–533, 2003.

[12] J. Luengo, S. García, and F. Herrera, On the choice of the best imputation methods for missing values considering three groups of classification methods, Vol. 32, No. 1. 2012.

[13] J. Alcalá-Fdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," Soft Comput., Vol. 13, no. 3, pp. 307–318, 2009.

[14] K. Baclawski, "Multiple Imputation by Chained Equations," vol. 30, no. April, pp. 1–3, 2011.

[15] N.S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," Am. Stat., Vol. 46, No. 3, pp. 175–185, 1992.

[16] A.K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognit. Lett., Vol. 31, No. 8, pp. 651–666, 2010.

[17] L. Beretta and A. Santaniello, "Nearest neighbor imputation algorithms: A critical evaluation," BMC Med. Inform. Decis. Mak., Vol. 16, Suppl 3, 2016.

[18] X.H. Cao, I. Stojkovic, and Z. Obradovic, "A robust data scaling algorithm to improve classification accuracies in biomedical data," BMC Bioinformatics, Vol. 17, No. 1, pp. 1–10, 2016.

[19] A. Idri, H. Benhar, J.L. Fernández-Alemán, and I. Kadi, "A systematic map of medical data preprocessing in knowledge discovery," Comput. Methods Programs Biomed., Vol. 162, pp. 69–85, 2018.

[20] H. Liu, "Discretization: An Enabling Technique," Data Min. Knowl. Discov., Vol. 6, pp. 393–423, 2002.

[21] J.R. Quinlan, "Improved Use of Continuous Attributes in C4.5," J. Artif. Int. Res., Vol. 4, No. 1, pp. 77–90, Mar. 1996.

[22] Y. Yang and G.I. Webb, "Discretization for naive-Bayes learning: Managing discretization bias and variance," Mach. Learn., Vol. 74, No. 1, pp. 39–74, 2009.

[23] Share, P.C. Analysis, and G.H. Dunteman, "Principal Components Analysis," in Principal Components Analysis, SAGE Publications, Inc, 1989, p. 96.

[24] C.W.M. Jae-On Kim, Factor Analysis: Statistical Methods and Practical Issues, 14th ed. SAGE Publications, Inc, 1978.

[25] P.R. Peres-Neto, D.A. Jackson, and K.M. Somers, "How many principal components? stopping rules for determining the number of non-trivial axes revisited," Comput. Stat. Data Anal., Vol. 49, No. 4, pp. 974–997, 2005.

[26] N. Poolsawad, L. Moore, C. Kambhampati, and J.G.F. Cleland, "Issues in the Mining of Heart Failure Datasets," Int. J. Autom. Comput., vol. 11, no. 2, pp. 162–179, Apr. 2014.

[27] thesis, "Correlation-based Feature Selection forMachine Learning," no. April, 1999.

[28] I. Guyon, "An Introduction to Variable and Feature Selection Isabelle," J. ofMachine Learn. Res. 3 1157-1182, Vol. 3, pp. 1157–1182, 2003.

[29] A.L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," Amficial Intell., Vol. 17, no. 1–2, pp. 245–271, 1997.

[30] R.K.A and L. George H. John b, "Wrappers for feature subset selection," Artif. Intell., Vol. 97, No. 1–2, pp. 273–324, 1997.

[31] A. Hapfelmeier and K. Ulm, "A new variable selection approach using Random Forests," Comput. Stat. Data Anal., Vol. 60, No. 1, pp. 50–69, 2013.

[32] V. Sigillito, "Pima Indians Diabetes Database," National Institute of Diabetes and Digestive and Kidney Diseases, 1990. [Online]. Available: http://ftp.ics.uci.edu/pub/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.names.

[33] "PIMA INDIAN DIABETES DATASET," UCI Machine Learning Repository, 1988. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes. [Accessed: 15-Apr-2018].