Received: April 12, 2021. Revised: June 17, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Recursive Parallel Partition Random Forest for Medical Disease Classification

Yosepu Cooli[1]* **Chitraivel Mahesh[2]**

[1]*Department of Computer Science and Engineering,*
Veltech Rangarajan Dr.Sagunthala R&D Institutute of Science and Technology, Chennai, India
[2]*Department of Information Technology,*
Veltech Rangarajan Dr.Sagunthala R&D Institutute of Science and Technology, Chennai, India
* Corresponding author's Email: cyosepu@gmail.com

Abstract: The development of computing technology has led to enormous improvement in the medical data and machine learning processing techniques. The increasing data usability in various field leads to the term big data, which is required in various applications such as medical, finance and so on. The existing models such as Random Forest, Support Vector Machine (SVM) and Decision tree have low capacity in handling the large volume of the data. In this research, the Recursive – Parallel Random Forest is proposed to effectively classify the large volume of medical data. The recursive partition method has been used to segment the input data and process the data in segmented manner. The recursive partition method segment the data and subset features based on data centric to improve the learning and provide adaptive gain ratio value to build Random Forest. The recursive partition method provides the data with high correlation that supports the random forest to perform the Random Forest parallel. The partition the data in data centric and balance the classes that helps the proposed model to handle the imbalance data effectively. The Parallel Random Forest method has been used to improve the performance of the medical data classification and the performance of Recursive partition- Parallel Random Forest (R-PRF) method was estimated using three UCI medical dataset such as Chronic Kidney Disease (CKD), Heart Disease and Diabetes. The standard classifiers and state-of-art method has been used to compare with the proposed R-PRF method and the experimental result shows that the R-PRF has the accuracy of 92.01 % and the existing Improved SVM radial method has 89.9 % accuracy.

Keywords: Big data, Parallel random forest, Recursive partition method, Standard classifiers, UCI medical datasets.

1. Introduction

Recently, the researchers have witness Big data changes on the complexities, definitions and future direction of the real world optimization problems [1]. Big data has been highly used in various areas such as medical, finance, etc.., and big data models have been developed to handle and process a large amount of data. MapReduce model is one of the data mining techniques and is widely used to effectively classify big data [2, 3]. The data analysis process is the crucial step in the many models of data mining and this process involves in several tasks such as data pre-processing, data extraction and data selection that helps in decision-making in getting best solution for the specified problem [4]. Development of transmission technology and information collection system tends to increases the amount of data. The conventional machine learning methods have lower efficiency in handling the big data analysis [5].

Traditional parallel algorithm is improved with helps of association rule generation algorithm and Map Reduce model with parallel optimization scheme of association rule algorithm to effectively classify the data [6]. Traditional single machine computation is not efficient in handling the big data, so multi-machine computation method is used to store and process the data in the distributed manner [7]. Machine learning models with the capacity of handling large-scale data and speed of data-mining technique received more attention in academia and industry. Studies on parallel and distributed data mining technique based on cloud computation

platform have higher efficiency [8, 9] and the existing methods has the drawbacks of lower computational efficiency [10]. In this research, the Recursive-Parallel Random Forest (R-PRF) method is proposed for classify the large number of data effectively. The recursive partition method segments the data in data-centric manner and adaptively build the Random forest. The partition of the data with high correlation and partition of data instance to the classes helps the proposed model to handle the imbalance data. The R-PRF method performance was estimated using three UCI medical dataset and four metrics were used to measure the efficiency of the R-PRF method.

The paper is organized as literature survey of the parallel data classification and medical data classification method is given in Section 2, the explanation about the recursive partition, parallel random forest is provided in Section 3 and the result of proposed R-PRF method is given in Section 4 and conclusion is given in Section 5.

2. Literature survey

Most of the data processing techniques have the high performance in the small and low dimensional data and existing methods have less effectiveness in large scale data. Some of the recent methods involves in applying multi-threading technique to improve the processing performance.

Barba-González [11] developed jMetalSP method that was the combination of multi-objective optimization of jMetal framework and the streaming facility of Apache Spark cluster computing system. Multi-objective metaheuristic can be easily adopted to the dynamic optimization problem of the multi-streaming data sources. The framework has the effective performance in the dynamic big data optimization method. The additional data sources with more realistic problems was not addressed by considering other optimization algorithms for big data optimization.

Wang [12] proposed hybrid multi-objective Firefly Algorithm for big data optimization. The six objective and six multi-objective problems was considered in the method to evaluate the performance. The obtained result shows that the Firefly method has the higher efficiency compared to existing method in big data optimization. The computational complexity of the method was high due to the combination of the developed method.

Ahmad [13] presented a parallel processing MapReduce method to enhance the performance of the medical data classification. A four-tier architecture was proposed involves in the input data, remove unnecessary data and analyze the data in parallel manner. The proposed method was implemented in Hadoop and MapReduce to estimate its overall performance. The result shows that the parallel processing MapReduce method has the higher performance in terms of effectiveness and it needs to be tested on the large dataset.

Wang [14] proposed available memory model to accurately capture the information by sensing the dependencies in the data. Based on this method, the Dependency Aware Storage Selection Mechanism has been developed for Spark to make dynamic and fine-grained storage decision. The developed method was evaluated in the garbage collection and shows higher performance in the computation. The performance of processing method needs to be increased for the developed method.

Huang [15] proposed three guiding principles to elaborate the process Gene Expression Programming (GEP) based on the analysis of GEP schema theory. The developed method analyzes the gene structure data in parallel and the input data size was considered in segment. The two datasets such as power system dataset and particle physical dataset were used to estimate the efficiency of the GEP method and the result shows that the speed of the GEP was improved significantly without affecting the accuracy. The scalability of the GEP method was high in dealing with big data with more number of CPU. When processing the integrated data block, the GEP based method has lower performance.

Harimoorthy [16] proposed improved SVM-Radial bias kernel method for the classification of medical data. The recursive feature selection method has been used to increases the performance and the SVM-Radial bias kernel was applied for the classification process in the method. The result show that the improved SVM method has the higher efficiency than existing method. The developed method requires more computation process for the data classification.

To overcome limitation of existing methods, the Parallel Random Forest method is proposed and it increases the speed of the process without affecting the performance.

3. Proposed method

Effective data classification model is required for the medical data classification and also requires to effectively handle the large amount of data. In this research, the Recursive-Parallel Random Forest (R-PRF) method is proposed to handle more number of data effectively. The three UCI medical data were used to estimate the performance of the developed method. The recursive partition method is applied to

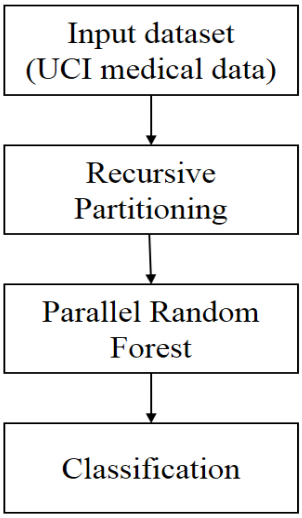

Figure. 1 The overview of the recursive-parallel random forest

segment the input data. The partition data is applied to the Parallel Random Forest method to classify the data. The proposed R-PRF method is compared with existing method in the medical data classification. The block diagram of proposed R-PRF method is shown in the Fig. 1.

3.1. Recursive partitioning

Decision Tree' recursive partitioning method was implemented in the ECL-ML Library for data analysis [17]. Recursion method is transform into iteration and ECL data-centric feature give details about few data structures to clearly explain the process. The Decision Tree (DT) Discrete Learning process of Gini Impurity/Info Gain was carried out to simplify the data. Two approaches of discrete and continuous version are implemented.

Hunt's Concept Learning System Framework is developed to build a decision tree and this is used to implement Recursive partition method. DT algorithm can be implemented based on this Framework and differ in the choice of the split/partition function.

Split Selection method S is developed based on BuildTree function and evaluated based on Training database D and if node t is not a leaf node, then a splitting criterion is applied. BuildTree function is recursively applied to each partition for continuously partitions the D database into t children nodes until stop criteria is reached. If t node is a leaf node, then stop criteria is reached. The DT method is consists of all returned leaf and split nodes (children nodes of t).

3.2. Parallel random forest on spark

Parallel Random Forest (PRF) algorithm on Spark [18] is proposed in this research to workload imbalance problems of large-scale data in a parallel and distributed manner, and mitigate the data communication cost. A hybrid parallel method combines data-parallel and task-parallel optimization method in PRF algorithm. In data-parallel optimization, a vertical data-partitioning method and a data-multiplexing method are applied. These two methods reduce the number of data transmission operations in distributed environment without affecting efficiency of the algorithm. In the task-parallel optimization, a dual-parallel approach is carried out in PRF algorithm training process and DAG task is created based on the RDD objects dependence. In the DAG, different task schedulers are applied to perform the tasks. The dual-parallel training method increases the PRF parallelization and increases the PRF performance. Task schedulers minimize the data communication cost among the Spark cluster and achieve better workload balance.

3.3. Random forest

RF was proposed as a combination of decision trees [19, 20] and this combination reduces the error in classification tasks. RF is a supervised and simple (ensemble of decision trees) method, which is fast and robust to the noise of the target data. The main idea of RF is to reduce the error of the prediction taking into account the decision trees included within the forest and the correlation among their predictions. The recursive features and parallel random forest is shown in Fig. 2.

In the Fig. 2, the recursive parallel random forest selects and segment the data based on data centric. The input data from datasets are consider as $x_1, x_2, x_3, \ldots, x_n$ and present in feature subsets. The segmented data is applied in the parallel random forest to perform classification. The segmented data is denoted as D_1, D_2, …, D_N. The data present in same partition have high correlation and this improve the performance of Decision tree. Focusing on one tree of the forest, let $P_i \in \mathbb{R}^{M_i \times N_i}$ where the i defines the i^{th} partition of samples (M_i) and features (N_i). P_i is randomly selected from the original data ($X \in \mathbb{R}^{M \times N}$) by generating random samples with replacement (i.e. by bootstrap).

At each node, the feature belonging to the subset N_i are considered candidates to split the available samples (M_i). The Gini Index (GI) is used to find the best splitting feature and cutoff point, as shown in Eq. (1).

$$G = \sum_{k=1}^{K} D_k(1 - D_k) \quad (1)$$

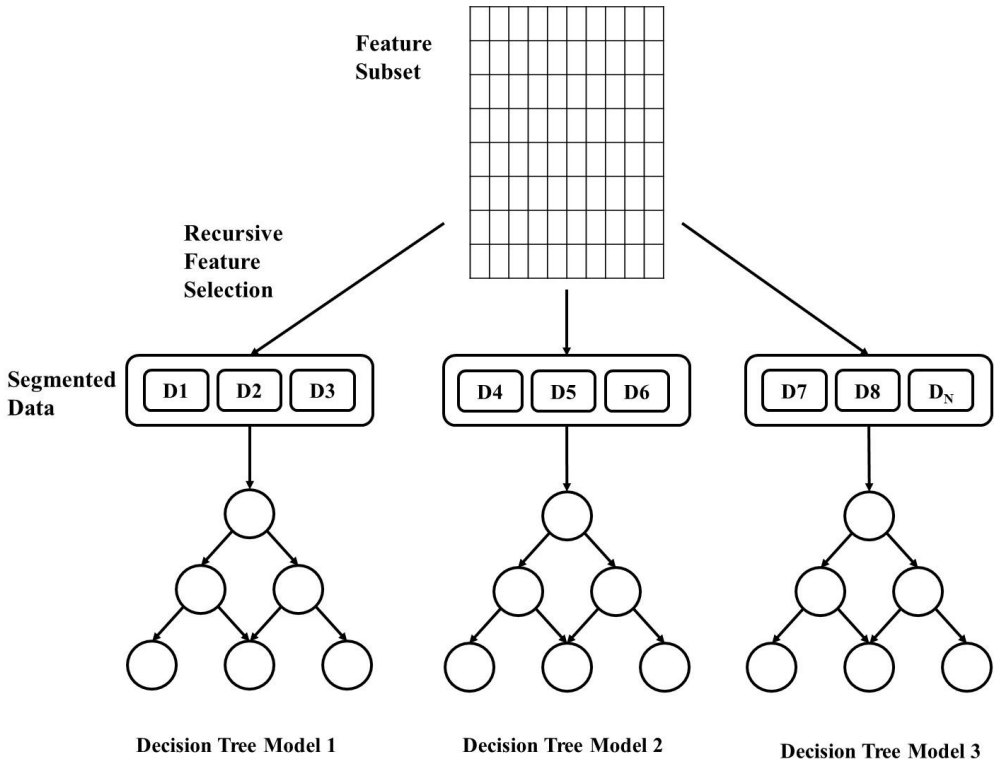

Figure. 2 The overview of the recursive-parallel random forest

Samples that have higher values than the cutoff point for the selected feature are directed to the right node (v_R) otherwise, they go to the left node (v_L). After several splittings, samples have moved from the root node (v_n) to the terminal nodes, also known as a terminal leaf which supply the predictions of the samples. The ensemble prediction ($\hat{Y} \in \mathbb{R}^{M\times 1}$) given by a forest is obtained as a combination of the results of the individuals trees; typically using the majority vote rule for classification is given in Eq. (6). The decision tree calculation is denoted as in Eq. (2).

$$ni_j = w_j G_j - w_{left(j)} G_{left(j)} - w_{right(j)} G_{right(j)} \quad (2)$$

where child node from right split on node j is denoted as $right(j)$, the node j child node from left is denoted as $left(j)$, node j Gini impurity value is denoted as G_j, reaching node j weighted number of samples is denoted as w_j, and the importance of node j is denoted as ni_j. Decision tree each features importance is calculated using Eq. (3).

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k\in all\ nodes} ni_k} \quad (3)$$

where N_{trees} is the total number of trees used in the RF.

The sum of all feature importance values is divided to normalize value between 0 and 1, as shown in Eq. (4).

$$normfi_i = \frac{fi_i}{\sum_{j\in allfeatures} fi_j} \quad (4)$$

Each tree's sum of feature importance value is calculated using Eq. (5).

$$RFfi_i = \frac{\sum_{j\in all\ trees} normfi_{i,j}}{T} \quad (5)$$

where T is the total number of trees, i in tree j normalized feature importance is denoted as $normfi_{i,j}$, and the feature importance.

$$Classification: \hat{Y}_i = mode_{n=1\ldots N_{trees}} \hat{Y}_n \quad (6)$$

where N_{trees} is the total number of trees used in the RF.

Two parameters require attention when optimizing a RF: the number of features that will be considered as split candidates (i.e. the size of the N_i subset), and the number of trees in the ensemble (i.e. N_{trees}). The former is often fixed by $sqrt(N)$ for classification or $N/3$ for regression, where N is the number of features in X. The latter is typically set equal to a few hundreds of trees because more trees do not necessarily lead to a better performance and just slow down the processing time.

Algorithm: Parallel Random Forest

Input:

k: the number of decision trees

T_{DSI}: The Data Sample Index

L_{FS}: a list of indexes of each feature subset's RDD object and the allocated slave nodes.

Output:

$PRF_{trained}$: the trained Parallel Random Forest model

1. For $i = 0$ to $(k - 1)$ do
2. For $j = 0$ to $(M - 2)$ do
3. Load feature subset $RDD_{FSj} \rightarrow loadData(L_{FS}[i])$;
4. $RDD_{GR,best} \rightarrow sc.parallelize(RDD_{FSj}).map$
5. Load sampled data $RDD_{i,j} \rightarrow (T_{DSI}[i], RDD_{FSj})$;
6. Calculate the gain ratio $GR_{i,j} \leftarrow GR(RDD_{i,j})$;
7. End map
8. $RDD_{GR,best}.collect().sortByKey(GR).top($
9. For each value $y(j, v)$ in $RDD_{GR,best}$ do
10. Split tree node $Node_j \leftarrow < y_{j,v}, value >$;
11. Append $Node_j$ to T_i;
12. End for
13. End for
14. $PRF_{trained} \leftarrow T_i$;
15. End for
16. Return $PRF_{trained}$

The algorithm of Parallel Random Forest describes the training of classifier based on input data. The number of decision tree is denoted as k and number of available data is denoted as M. The iteration of $k - 1$ is i and the iteration of $M - 2$ is j. The L_{FS} is a list of index of each feature subset and this is loaded in Parallel Random Forest and the sample index of data $T_{DSI}[i]$ is loaded in the model. The Gain Ratio GR is measured in feature subset and build the tree. The feature subset is sorted based on gain ratio of the features. Nodes and labels $y_{j,v}$ were obtained from the input dataset. The trees are split from the feature subset based on the gain ratio and continue the process until the number of decision tree. This provides the trained Random Forest and test data is applied for classification. The classification is performed based on Eq. (6) using the trained Random Forest.

4. Experimental result

Most of the existing method such as Random Forest, SVM and Decision tree methods have the limitations of lower efficiency to process huge amount of data. This research involves in applying R-PRF method to process the large amount of data without affecting the accuracy of the classification. This section provides the detailed description about the performance of the developed R-PRF method. The three UCI datasets such as Chronic Kidney Disease (CKD) [21], Heart Disease [22] and Diabetes [23] were used to evaluate the performance of the model. The CKD dataset has 400 data instances with 25 attributes. The Heart Disease dataset has 303 data instances with 75 attributes. The Diabetes dataset has 20 number of attributes for the classification. The evaluation metrics such as accuracy, precision, sensitivity and specificity are measured to analysis the performance. The metric formulas are shown in Eqs. (7) to (10).

$$Accuracy\ (\%) = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (7)$$

$$Precision\ (\%) = \frac{TP}{TP+FP} \times 100 \quad (8)$$

$$Sensitivity(\%) = \frac{TP}{TP+FN} \times 100 \quad (9)$$

$$Specificity\ (\%) = \frac{TN}{TN+FP} \times 100 \quad (10)$$

The proposed R-PRF method is implemented in the system consists of Intel i7 processor with 8 GB of RAM and 4 GB of Graphics Card. The training data is set as 80 % and testing data is set as 20 % in the classification process. The computational complexity of the R-PRF method is $O(N^2)$ due to the number of required elementary is less in proposed method. The Accuracy value of R-PRF method is estimated in the three UCI dataset, as shown in Fig. 2.

The R-PRF method is evaluated with accuracy in three datasets such as CKD, Heart disease and diabetes. The accuracy of the R-PRF method is high in three UCI datasets than the existing method, as shown in Fig. 3. Partitioning the input data helps to improve the performance of the developed method in medical dataset. The classifiers have lower

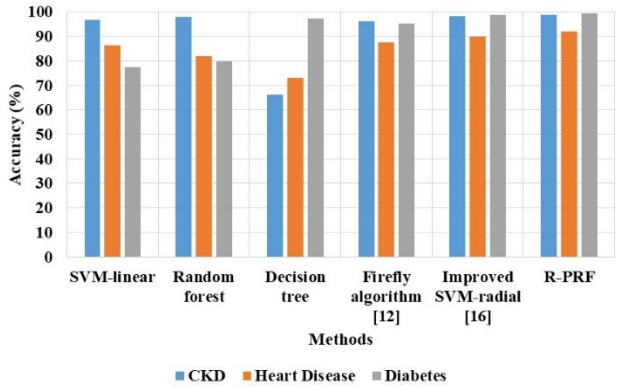

Figure. 3 Accuracy of the R-PRF method in three dataset

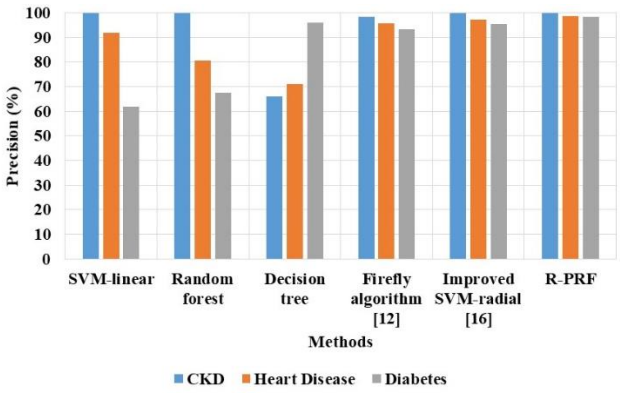

Figure. 4 Precision of R-PRF method in three dataset

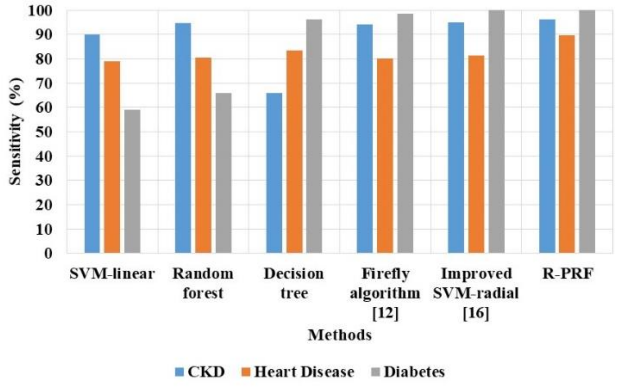

Figure. 5 Sensitivity of the R-PRF in three dataset

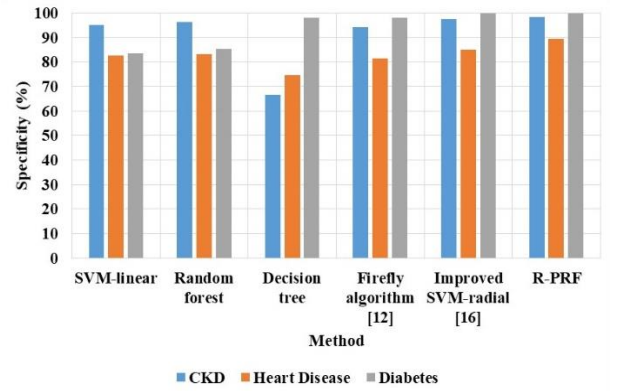

Figure. 6 The specificity of R-PRF method in three dataset

performance in heart disease dataset compared to CKD and diabetes datasets due to heart disease dataset has more number of attributes. The R-PRF method has 18% higher accuracy compared to the traditional RF. Although, the R-PRF method outperforms the SVM, decision tree and Improved SVM-radial [16] classifier. The decision tree method has lower in increases the number of instance and has lower performance in CKD. The proposed R-PRF method partition the data based on data centric manner to adaptively build the Random Forest. The proposed R-PRF method segment the data with high correlation that supports the classification in parallel manner. The accuracy of the R-PRF method is achieved as 92.01 % in the Heart disease dataset, while existing method has 89.9 % accuracy.

The precision value of R-PRF method is estimated in the three UCI dataset, as shown in Fig. 4. The R-PRF method partition the data instance and features in the class adaptively and train the Random Forest to solve the imbalance data problem. The R-PRF has the higher precision value due to the data partition method improve the performance of RF.

The R-PRF method achieves the accuracy of 100 %, 98.56 %, and 98.38 % in the CKD, Hearth disease and Diabetes dataset, respectively. The R-PRF method significantly improve the performance in the classification of medical data compared to RF method. The traditional methods such as SVM, RF, decision tree has less performance compared to R-PRF method. The R-PRF method outperforms the state-of-art method in the data classification. The precision value of R-PRF method is achieved as 98.56 %, while standard SVM method has 91.9 % precision. The sensitivity of the R-PRF method is evaluated in three datasets, as shown in the Fig. 5. The sensitivity of the R-PRF method is higher compared to the standard classification method and existing method. The SVM, Improve SVM-radial, RF and decision tree standard classifiers were used to compare with the proposed R-PRF method. The R-PRF has the higher sensitivity due to the partition method applied in the input data. The sensitivity of R-PRF method 89.56 %, while existing Improved SVM-radial method achieved 81.4 % sensitivity. The parallel RF method has the higher efficiency compared to traditional RF. The sensitivity of R-PRF method and improved SVM-radial method has achieved 100 % sensitivity. The Recursive partition method segment the data based on data centric manner to adaptively build the random forest. The segment the data with high correlation that supports the random forest to classify in parallel manner. The R-PRF method is evaluated in terms of specificity and compared with the other traditional classifier and

Table 1. Performance analysis of R-PRF method in UCI medical data

Parameters	Disease	SVM-linear	Random forest	Decision tree	Firefly algorithm [12]	Improved SVM-radial [16]	R-PRF
Accuracy	CKD	96.7	97.8	66.3	96.2	98.3	98.92
	Heart Disease	86.5	82	73	87.5	89.9	92.01
	Diabetes	77.6	79.9	97.4	95.2	98.7	99.5
Precision	CKD	100	100	65.9	98.4	100	100
	Heart Disease	91.9	80.5	71.1	95.6	97.2	98.56
	Diabetes	61.9	67.4	96.1	93.2	95.5	98.38
Sensitivity	CKD	90	94.7	65.9	94.2	95	96.18
	Heart Disease	79.1	80.5	83.3	80.2	81.4	89.56
	Diabetes	59.1	66	96.1	98.7	100	100
Specificity	CKD	95.2	96.3	66.7	94.3	97.6	98.42
	Heart Disease	82.7	83.3	74.5	81.3	84.9	89.38
	Diabetes	83.6	85.2	98.1	98.2	100	100

state-of-art method, as given in Fig. 6. The R-PRF has the higher specificity than the other standard classifier. The partitioning of the input dataset improves the performance of the classification in medical datasets. The R-PRF method segment the data instance in the classes that helps to handle the imbalance data. The parallel processing in the RF increases the efficiency in the classification. The R-PRF and the improved SVM-radial method has the higher efficiency in the diabetes dataset. Overall, the R-PRF method has the higher specificity in three UCI datasets than standard classifiers and existing Improved SVM-radial method. The R-PRF method has the specificity of 89.38 %, while existing improved SVM-radial method has 84.9% specificity.

Various evaluation metrics were used to estimate the efficiency of R-PRF method, as shown in Table 1. The three UCI medical datasets were applied to evaluate the efficiency of the R-PRF and compared with the existing method. The comparative analysis shows that the R-PRF has the higher efficiency than existing method. The R-PRF method segment the data in data centric manner and adaptively train the random forest. The R-PRF has the higher efficiency due to partition of the input data and RF parallel processing. The R-PRF method and existing improved SVM-radial method has the higher efficiency in the Diabetes dataset than other datasets. The R-PRF method has the accuracy of 99.5 % in diabetes datasets, while existing method has 98.7 % accuracy.

The R-PRF method has the higher efficiency in three UCI dataset than other existing method. The R-PRF has the capacity to handle more number of data with efficiency.

5. Conclusion

Medical data processing based on machine learning is widely applied in the big data for disease classification. In this research, the R-PRF method is proposed to increases the classification efficiency in large volume of data. The recursive partition algorithm is applied to segment the input data for effective classification. The proposed R-PRF method has the advantages of segment the data in data-centric manner to adaptive train the Random Forest. The parallel RF method has been applied to effectively process the large number of data effectively. The partition of the input data and parallel processing of the random forest improves the efficiency of classification. The three UCI datasets such as CKD, Heart Disease and Diabetes were used to estimate the performance of the R-PRF method. The proposed R-PRF method outperforms the existing models such as SVM, Improved SVM-radial and RF. The proposed R-PRF method has the capacity to handle the imbalance data and adaptively classify the input data. The result shows that R-PRF has the precision of 98.38 %, while existing method has 95.5 % precision. In future, deep learning method can be applied to improve the performance of medical data classification.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper background work, conceptualization, methodology, dataset collection, implementation, result analysis and comparison, preparing and editing draft, visualization have been done by first author. The supervision, review of work and project administration, have been done by second author.

References

[1] S. Aslan and D. Karaboga, "A genetic Artificial Bee Colony algorithm for signal reconstruction based big data optimization", *Applied Soft Computing*, pp. 106053, 2020.

[2] C. Banchhor and N. Srinivasu, "Integrating Cuckoo Search-Grey wolf optimization and Correlative Naive Bayes classifier with Map Reduce model for big data classification", *Data & Knowledge Engineering*, pp. 101788, 2019.

[3] J. Chen, K. Li, H. Rong, K. Bilal, N. Yang, and K. Li, "A disease diagnosis and treatment recommendation system based on big data mining and cloud computing", *Information Sciences*, Vol. 435, pp. 124-149, 2018.

[4] M. Mohamad, A. Selamat, O. Krejcar, H. Fujita, and T. Wu, "An analysis on new hybrid parameter selection model performance over big data set", *Knowledge-Based Systems*, Vol. 192, pp. 105441, 2020.

[5] Y. Xu, H. Liu, and Z. Long, "A distributed computing framework for wind speed big data forecasting on Apache Spark", *Sustainable Energy Technologies and Assessments*, Vol. 37, pp. 100582, 2020.

[6] Y. Cao, P. Li, and Y. Zhang, "Parallel processing algorithm for railway signal fault diagnosis data based on cloud computing", *Future Generation Computer Systems*, Vol. 88, pp. 279-283, 2018.

[7] K. K. Sethi and D. Ramesh, "HFIM: a Spark-based hybrid frequent itemset mining algorithm for big data processing", *The Journal of Supercomputing*, Vol. 73, No. 8, pp. 3652-3668, 2017.

[8] S. T. Habib, and Z. Ansari, "An analysis of MapReduce efficiency in document clustering using parallel K-means algorithm", *Future Computing and Informatics*, Vol. 3, pp. 200-209, 2018.

[9] J. Maillo, S. Ramírez, I. Triguero, and F. Herrera, "kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data", *Knowledge-Based Systems*, Vol. 117, pp. 3-15, 2017.

[10] S. Ramírez-Gallego, S. García, J. M. Benítez, and F. Herrera, "A distributed evolutionary multivariate discretizer for big data processing on apache spark", *Swarm and Evolutionary Computation*, Vol. 38, pp. 240-250, 2018.

[11] C. Barba-González, J. García-Nieto, A. J. Nebro, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, and J. F. Aldana-Montes, "jMetalSP: a framework for dynamic multi-objective big data optimization", *Applied Soft Computing*, Vol. 69, pp. 737-748, 2018.

[12] H. Wang, W. Wang, L. Cui, H. Sun, J. Zhao, Y. Wang, and Y. Xue, "A hybrid multi-objective firefly algorithm for big data optimization", *Applied Soft Computing*, Vol. 69, pp. 806-815, 2018.

[13] A. Ahmad, A. Paul, S. Din, M. M. Rathore, G. S. Choi, and G. Jeon, "Multilevel data processing using parallel algorithms for analyzing big data in high-performance computing", *International Journal of Parallel Programming*, Vol. 46, No. 3, pp. 508-527, 2018.

[14] B. Wang, J. Tang, R. Zhang, W. Ding, and D. Qi, "A dependency-aware storage schema selection mechanism for in-memory big data computing frameworks", *International Journal of Parallel Programming*, Vol. 47, No. 3, pp. 502-519, 2019.

[15] Z. Huang, M. Li, C. Chousidis, A. Mousavi, and C. Jiang, "Schema Theory-Based Data Engineering in Gene Expression Programming for Big Data Analytics", *IEEE Transactions on Evolutionary Computation*, Vol. 22, No. 5, pp.792-804, 2017.

[16] K. Harimoorthy and M. Thangavelu, "Multi-disease prediction model using improved SVM-radial bias technique in healthcare monitoring system", *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-9, 2020.

[17] V. M. Herrera, T. M. Khoshgoftaar, F. Villanustre, and B. Furht, "Random forest implementation and optimization for Big Data analytics on LexisNexis's high performance computing cluster platform", *Journal of Big Data*, Vol. 6, No. 1, pp. 68, 2019.

[18] J. Chen, K. Li, Z. Tang, K. Bilal, S. Yu, C. Weng, and K. Li, "A parallel random forest algorithm for big data in a spark cloud computing environment", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, No. 4, pp. 919-933, 2016.

[19] E. Izquierdo-Verdiguier and R. Zurita-Milla, "An evaluation of Guided Regularized Random Forest for classification and regression tasks in remote sensing", *International Journal of*

Applied Earth Observation and Geoinformation, Vol. 88, pp. 102051, 2020.
[20] G. Chen, Q. Li, F. Shi, I. Rekik, L. Wang, and Z. Pan, "RFDCR: Automated brain lesion segmentation using cascaded random forests with dense conditional random fields", *NeuroImage*, pp. 116620, 2020.
[21] A. Khamparia, G. Saini, B. Pandey, S. Tiwari, D. Gupta, and A. Khanna, "Chronic kidney disease classification with multimedia data learning using deep stacked autoencoder network", *Multimedia Tools and Applications*, pp.1-16.
[22] https://archive.ics.uci.edu/ml/datasets/heart+disease
[23] https://archive.ics.uci.edu/ml/datasets/diabetes