Data Preprocessing With K Nearest Neighbor, Normalization In Big Data

K. Sathesh Kumar[1]*, S. Ramkumar[2], P.Nagaraja[3], A. Robert Singh[4]

[1, 2,3,4] School of Computing, Kalasalingam Academy of Research and Education,*

Krishnankoil, Virudhunagar, Tamil Nadu, India

Abstract

The proposed MapReduce framework works more effectively in big data. The big data concern large-volume, complex, growing data sets with multiple, autonomous sources. This may lead problems such as noisy data, missing data and redundant data because the big data is collected from multiple systems or sources which affects the decision making for data mining with big data in this work the k nearest neighbor algorithm is used to calculate the missing values. Assume each feature of training data sets has a distinct dimension in some scope, and take an observation value for its feature coordinates in that dimension, then acquire some set of points from space. Then assume the identical of two points would be the distance between them in this space depends on some metric. This method is used to gain proper dataset and it increases data quality for further processes.

Keywords: *Big data, Imputation, Data Mining and Classification*

1. Background

1.1 Introduction

In this work such unnecessary [4] data, redundant data and noisy data in dataset are handled by using K nearest neighbor algorithm. The dataset contain nominal and continuous values. In the transformation step first convert the continuous data like duration, src_bytes, dst_bytes, wrong_fragment, urgent are converted into discrete form and the nominal values like http, UDP, TCP are converted into numerical values for imputation of missing values and noise cleansing. Now the dataset contain the numerical values but there is missing values for some attributes. Those missing values are imputed using K nearest neighbor algorithm. K nearest neighbor algorithm found nearest neighbor and calculate distance between the neighbor values and based on k value calculate the mean to fill the missing values [6-8]. The same algorithm is used in noise cleansing. The data points in the training set are divided into outliers, prototype and absorbed points. After examining the training data set, deletes each point when it cannot associate with a correct class, then it is termed as outliers. Otherwise, if it is associated with a correct class, it placed to the dataset. Then pick any data point from the dataset to recognize it is a correct class using value k. Then check if it any prototype is added to the dataset if it is not added then it is considered as a pure dataset. Hence the noisy data are removed from the dataset. This method is used to gain proper dataset and it increases data quality for further processes [11-13].

1.2 Big data Processing

Real world data could be dirty and it leads extraction of useless patterns or rule. This is mainly due to lacking of attribute values, containing errors or outliers and including discrepancies in the dataset [14-16]. The data preprocessing is the process of creating small set of data than the original dataset size it allows to improve efficiency in the data mining process and it is used to get quality data from the huge volume of data which obtain quality patterns or rules from the reduced dataset. In this proposed work the data pre processing is handled by transformation, filling missing values and noise removal [5].

1.3 Transformation

In this work KDD cup dataset, lipid profile dataset are used that contains different types of data values such as categorical value, nominal value and continuous values. The main objective of data transformation process is to transform data in the best way possible to the application of Data Mining [5]

1.3.1 Min-Max Normalization

In this transformation process convert the continuous values into defined range of values called linear transformation.[6] Suppose in the dataset an attribute value ranges from 14000 to 20000 transformation function is used to map the attribute to the range from 0 to 1.

Min-max normalization: $[mini_{data}, maxi_{data}] \rightarrow [\, new_{mini_{data}}, new_{maxi_{data}} \,]$

$$v' = \frac{v - mini_{data}}{mini_{data} - maxi_{data}} (new_{mini_{data}} - new_{maxi_{data}}) + new_{mini_{data}} \qquad (1)$$

Where v^' is the min max normalized value. The above equation is used to transform the continuous data values into defined range of values in dataset.

1.3.2 Z-Score Normalization

The technique which offers the normalized range or values of data from the unstructured original data with the help of concepts like mean and standard deviation then the technique is called as Z-score Normalization. [6] So the Z-score parameter is used to convert from unstructured data to structured data, by using following formula:

$$v' = \frac{v - mean_{data}}{standard - deviation_{\ data}} \qquad (2)$$

$$mean_{data} = \frac{1}{n} \sum_{i=1}^{n} v_i \qquad (3)$$

$$standard - deviation_{\ data} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (v_i - mean_{data})^2} \qquad (4)$$

where v' is the Z-score normalized value and v_i is the value of the row in ith column. In this method, the data set has five rows such as A, B, C, D and E with distinct variables or columns that are 'm' in each row. From each and every row the normalized data will be identified through the z-score method. Assume if few rows have the similar attribute values; the standard deviation of the row is zero after that all values of that row are same to zero. *It* is similar to the Z Min-Max Normalization

1.3.3 Decimal scaling normalization

The decimal scaling normalization method gives the limit between -1 and 1.The decimal scaling normalization is evaluated by using following formula:

$$v' = \frac{v}{10^j} \qquad (5)$$

where v is the range of values and j is the smallest integer max ($|v_i|$) <1

From the above 1,2,3,4 and 5 equation the continuous values are transformed into define range of values which is widely used for the purpose of data mining. After transforming the dataset values, next step of data pre processing is missing value imputation and noise removal which plays an important role in data mining.

2. Proposed Work

2.1 Missing Value Imputation

Missing data is the state where some values of some attributes are missing [17-19]. This is not uncommon. Handling dataset with missing data is time consuming. Fixing up problems happened by missing data usually takes longer than the analysis itself. Finding the missing values is the best solution but this is not always possible. There are several reasons for missing values in the dataset. If the missing data is non-informative then a fix is easier. For example, if a data point is missed because it was huge then this could cause some partiality and a simple fix is not possible.[7] There are different fix-up methods are used to find the missing values of non informative reasons:

- The simple solution for missing values is deleting the case with missing data. This is OK if this only causes the loss of a fairly small number of cases.
- Fill-in or impute the missing values. With the help available data in the dataset predict the missing values. One of the easy methods to predict the missing value in the dataset is simply replacing the missing value of an interpreter with the average value of that interpreter. Another way is to use regression on the other interpreters. It's not clear how much the investigative and deduction on the filled-in dataset is affected. Some additional uncertainty is developed by the imputation which needs to be tolerated for.
- Missing inspection correlation. Consider a pair of data with some observations missing. The missing values are imputed using mean and standard deviation and mean between pair of data it imputes the missing value even when a member of a pair is missing. An analogous method is available for regression problems.
- Another method is maximum likelihood methods can be used considering the multivariate normality of the data. The Expectation Maximization algorithm is often used for maximum likelihood algorithm.

In this work the k nearest neighbor algorithm is used to calculate the missing values. Assume each feature of training data sets has a distinct dimension in some scope, and take an observation value for its feature coordinates in that dimension, then acquire some set of points from space. Then assume the identical of two points would be the distance between them in this space depends on some metric. This is calculated by using following equation

$$E(x,y) = \sqrt{\sum_{i\epsilon D}(A_{xi} - B_{yi})^2}$$

A_xi and B_yi are the values of attribute i in cases x and y respectively and D is the set of attributes with missing values in both the cases.

An algorithm will decide which similar points will select from the training data set when select the class to predict a new observation. To predict a new observation, the data points get from the k closest and also pick some several general classes among these. So this is termed as a k nearest neighbor algorithm

Algorithm 4: Missing value imputation using K nearest neighbor algorithm

Input: dataset with missing values and positive integer

Output: dataset without missing values

```
1. A positive integer k is identified, along with a new sample
2. Select the k entries from the dataset which are closest to the new sample in a column
   using Euclidean distance formula
      From the k value find mean values
3. Fill the missing value with mean value
4.
```

Description

Algorithm 4, impute the missing values in the dataset using Euclidean distance. Initially the dataset is given as input to the K nearest neighbor algorithm then it selects the k entries from the dataset which are closest to missing value. Then based on k values mean of data are calculated. Finally mean value is the imputed missing value it calculate the missing values columnar wise simultaneously noise in the dataset is removed row wise which gives more effective and pure dataset for mining process.

2.2 Noise Removal

Removing data that contain noise is an important objective of data cleaning as noise hinders most types of data analysis. Noise defined as irrelevant or meaningless data. For most existing data cleaning methods, the focus is on the detection and removal of noise that is the result of an imperfect data collection process. The data points in the training set are divided into three types to remove the noise are outliers, prototypes and absorbed points. Outlier is a type which is considered as not be recognized as correct class and it is added to the database later then the prototypes are a type where the smallest set of points needed to correctly identify the other non-outliers points and finally absorbed points is a type which are not related to outliers. But it will recognize based on the prototype points, these three types of points are used to define the dataset (https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm). There are various techniques are available to remove the noisy data but K nearest neighbor algorithm is most effective and simple algorithm to handle noisy data in row wise. The process of noise removal in dataset using K nearest neighbor algorithm is given below:

Algorithm 5: Noise removal using K nearest neighbor algorithm
Input: Dataset with noise
Output: Dataset without noise

```
1. Analysis the training data set, removes each point and test whether it is recognized by
   correct class or not
2.     if so, recognized as correct class, then place it back in the set
3.     If so, not identified as correct class, and assumed as an outlier and
       must not be placed again
4. Create a database which doesn't exist, and include a random point
5. Choose any of the points from real data set to know whether it is recognized by the
   correct class based on the points in the new database, k nearest neighbor with k values are
   utilized
6.     If so, recognized as correct class, then it is called as absorbed point, and may
   remove from the new database
7. When it is recognized as a wrong class, it just removed from the data set and it will
   included to the new database prototypes
8. Continue this process using  the real data set  like these
9. Again execute steps 5 and 8 till when new prototypes are not included.
```

Description

The Algorithm 5, analysis the training dataset to check whether the data set is associated with correct class or not depend upon the types of data points. The data points are classified as three types outlier is a type which is considered as not be recognized as correct class and it is added to the database later then the prototypes are a type where the smallest set of points needed to correctly identify the other non-outliers points and finally absorbed points is a type which are not related to outliers. But it will recognize based on the prototype points. If the data point is associated with correct class, then the data point is placed into the data set which means it doesn’t contain noisy data. Otherwise it is considered as outlier that should not be put back in the dataset which data point is treated as noisy data and makes a new data base based on the data points and adds a random point. Then again choose a data point from original dataset and check the data point is associated with the correct class using K nearest neighbor algorithm. If the data point is associated with the correct class it is an absorbed point and it can be removed from new dataset. Otherwise the data point will be deleted from the real data set, and then finally the data point is included in the newly created database of prototype. It is a recursive process until no new prototypes are added to the database. From this algorithm the noise in the dataset are removed in row wise. Thus the algorithm 4, 5 works simultaneously to fill the missing values through column wise calculation of K Nearest Neighbor algorithm and remove the noisy data in row wise using the same K Nearest Neighbor algorithm.

3. Performance Evaluation

For the practical purpose, (Discussed in previous work [1-3] [15]) the proposed Content, computation and Network Aware (CCNA) MapReducer is compared with the existing Content Aware (CA) MapReducer and Content, computation Aware (CCA) MapReducer are compared in terms Root Mean Square Error, Normalised Root Mean Square Error and Mean Absolute Error with the KDD cup data set and Lipid profile dataset which is described in section 3.4. The impact of this work is to preprocess the by filling missing values in column wise and noise cleansing in row wise simultaneously and this both preprocessing techniques are processed based on K nearest neighbor algorithm. Hence this simultaneous process will reduce the time consumption of data preprocessing.

3.1 Root Mean Square Error

The root-mean-square error (RMSE) is a measure of the differences between values predicted by a model or an estimator and the values actually observed (https://en.wikipedia.org/wiki/Root-mean-square_deviation). It can be calculated by using following formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (X_{obs,i} - X_{model,i})^2}{n}}$$

where Xobs is observed values and Xmodel is modelled values at time/place i.

Table 3.1. Root Mean Square Error for KDD dataset

Input data size in MB	**RMSE**		
	CA MapReducer	**CCA MapReducer**	**CCNA MapReducer**
1000	0.322	0.320	0.312
2000	0.323	0.319	0.310
3000	0.322	0.317	0.308
4000	0.321	0.315	0.307
5000	0.316	0.314	0.304

From the figure 3.1 and table 3.1 shows that RMSE value against data size. The proposed CCNA MapReducer method is authenticated, that shows 0.304 RMSE value but the existing CA MapReducer shows 0.316 RMSE value and CCA MapReducer shows 0.314 RMSE value at the point of 5000 mb data size.

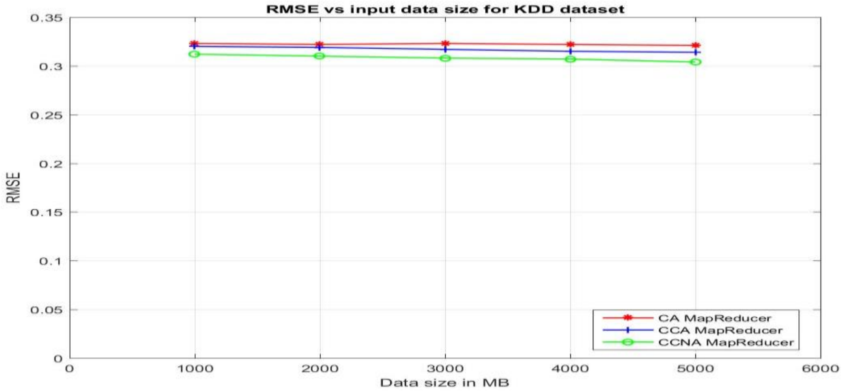

Figure 3.1 Root Mean Square Error for KDD Dataset

Whereas for 2000 mb data size the proposed CCNA MapReducer method shows 0.310 RMSE value but the existing CA MapReducer shows 0.323 RMSE value and CCA MapReducer shows 0.319 RMSE value for KDD dataset. From this it is found that the CCNA MapReducer with proposed data pre processing technique has less RMSE value than the other techniques.

Table 3.2 Root Mean Square Error for Lipid Profile Dataset

Input data size in MB	RMSE		
	CA MapReducer	CCA MapReducer	CCNA MapReducer
1000	0.315	0.313	0.303
2000	0.312	0.311	0.301
3000	0.311	0.307	0.299
4000	0.3112	0.306	0.297
5000	0.310	0.305	0.295

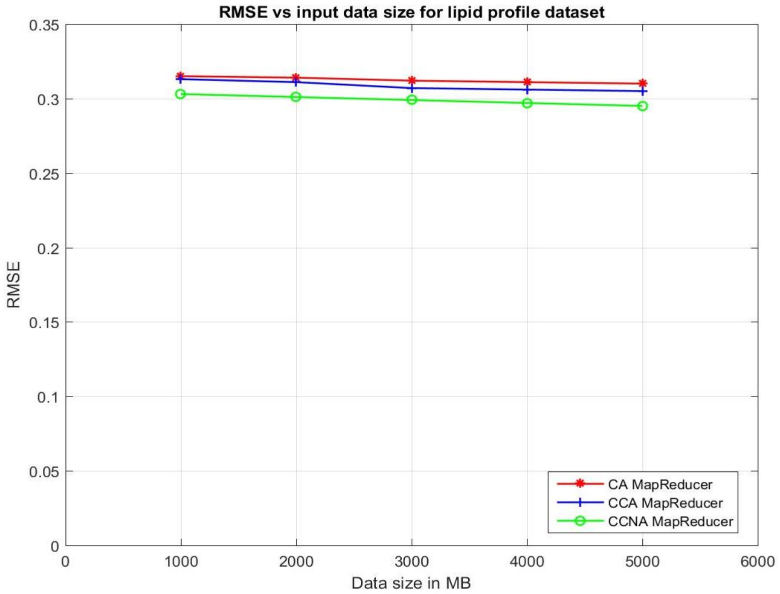

Figure 3.2. Root Mean Square Error for lipid profile dataset

From the figure 3.2 and table 3.2 shows that RMSE value against data size. The proposed CCNA MapReducer method is authenticated, that shows 0.295 RMSE value but the existing CA MapReducer shows 0.310 RMSE value and CCA MapReducer shows 0.305 RMSE value at the point of 5000 mb data size. Whereas for 2000 mb data size the proposed CCNA MapReducer method shows 0.301 RMSE value but the existing CA MapReducer shows 0.312 RMSE value and CCA MapReducer shows 0.311 RMSE value for lipid profile dataset. From this it is found that the CCNA MapReducer with proposed data pre processing technique has less RMSE value than the other techniques.

3.2 Normalised Root Mean Square Error

Non-dimensional forms of the RMSE are useful because often one wants to compare RMSE with different units. There are two techniques: normalize the RMSE to the range of the observed data, or normalize to the mean of the observed data.

$$NRMSE = \frac{RMSE}{X_{obs,\max} - X_{obs,\min}}$$

Table 3.3 Normalised Root Mean Square Error for KDD Dataset

Input data size in MB	RMSE		
	CA MapReducer	CCA MapReducer	CCNA MapReducer
1000	0.324	0.322	0.318
2000	0.322	0.321	0.313
3000	0.322	0.320	0.312
4000	0.321	0.318	0.308
5000	0.319	0.317	0.307

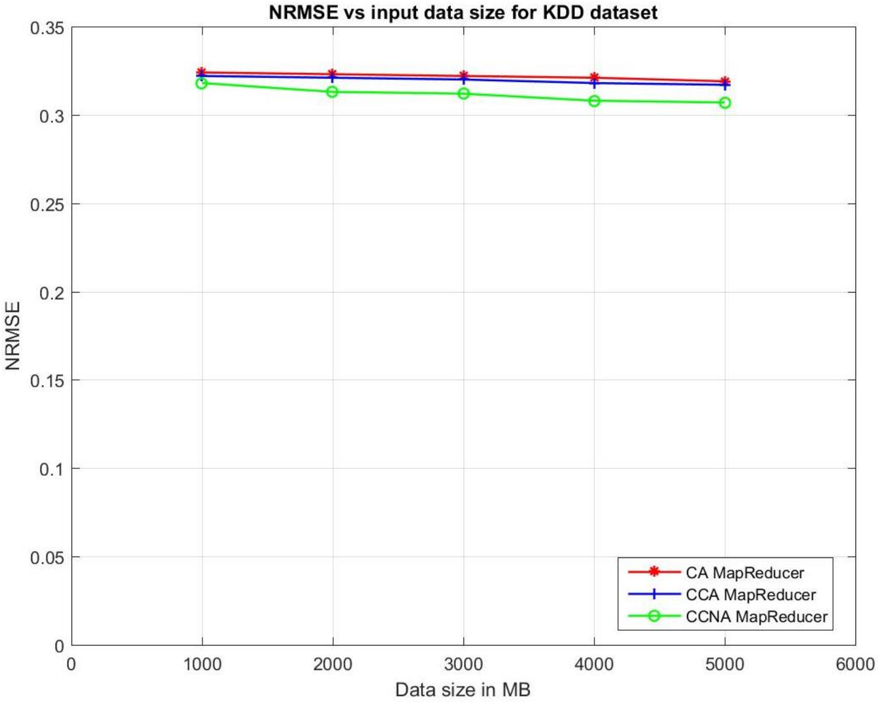

Figure 3.3 Normalised Root Mean Square Error for KDD Dataset

From the figure 3.3 and table 3.3 shows that NRMSE value against data size. The proposed CCNA MapReducer method is authenticated, that shows 0.307 NRMSE value but the exiting CA MapReducer shows 0.317 NRMSE value and CCA MapReducer shows 0.319 NRMSE value at the point of 5000 mb data size. At the point of 2000 mb data size the proposed CCNA MapReducer method shows 0.313 NRMSE value but the exiting CA MapReducer shows 0.322 NRMSE value and CCA MapReducer shows 0.321 NRMSE value for KDD dataset. From this it is found that the CCNA MapReducer with proposed data pre processing technique has less NRMSE value than the other techniques.

Table 3.4 Normalised Root Mean Square Error for Lipid Profile Dataset

Input data size in MB	RMSE		
	CA MapReducer	CCA MapReducer	CCNA MapReducer
1000	0.324	0.321	0.312
2000	0.323	0.319	0.309
3000	0.321	0.316	0.308
4000	0.320	0.315	0.307
5000	0.319	0.314	0.305

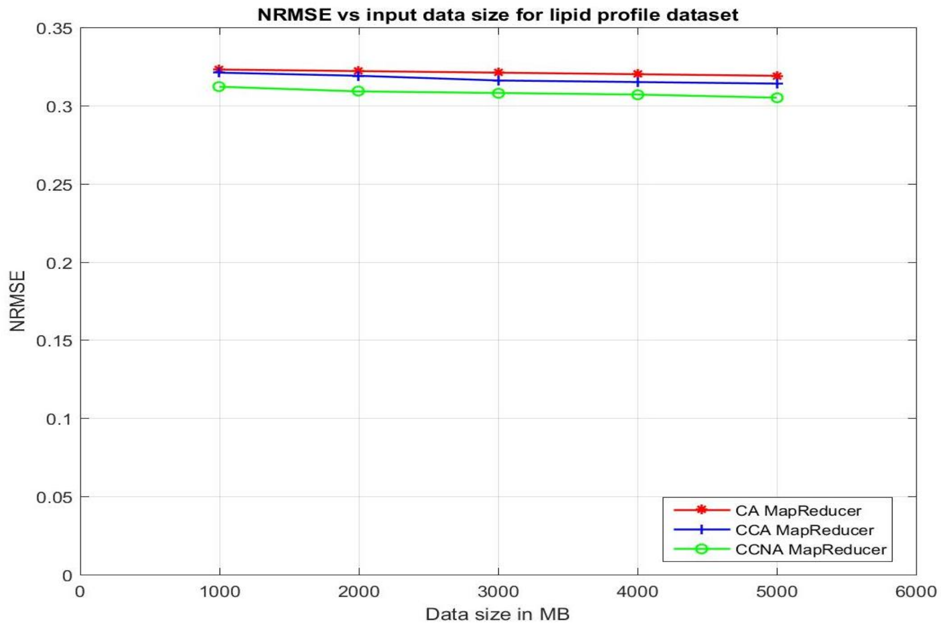

Figure 3.4 Normalised Root Mean Square Error for Lipid Profile Dataset

From the figure 3.4 and table 3.4 shows that NRMSE value against data size. The proposed CCNA MapReducer method is authenticated, that shows 0.305 NRMSE value but the exiting CA MapReducer shows 0.319 NRMSE value and CCA MapReducer shows 0.314 NRMSE value at the point of 5000 mb data size. At the point of 2000 mb data size the proposed CCNA MapReducer method shows 0.309 NRMSE value but the exiting CA MapReducer shows 0.323 NRMSE value and CCA MapReducer shows 0.319 NRMSE value for lipid profile dataset. From this it is found that the CCNA MapReducer with proposed data pre processing technique has less NRMSE value than the other techniques.

3.3 Mean Absolute Error

The mean absolute error (MAE) is a metric used to measure how close predictions are to the eventual outcomes. The mean absolute error is given by

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |p_i - q_i|$$

Where pi is the forecasted value and qi is the real value.

Table 3.5 Mean Absolute Error for KDD Dataset

Input data size in MB	RMSE		
	CA MapReducer	CCA MapReducer	CCNA MapReducer
1000	0.283	0.282	0.275
2000	0.282	0.281	0.274
3000	0.281	0.279	0.273
4000	0.280	0.278	0.272
5000	0.279	0.276	0.271

From the figure 3.5 and table 3.5 shows that MAE value against data size. The proposed CCNA MapReducer method is authenticated, that shows 0.271 MAE value but the existing CA MapReducer method shows 0.279 MAE value and CCA MapReducer method shows 0.276 MAE value at the point of 5000 mb data size. At the point of 2000 mb size the proposed CCNA MapReducer method shows 0.274 MAE value but the existing CA MapReducer method shows

0.282 MAE value and CCA MapReducer method shows 0.281 MAE value for KDD dataset. From this it is found that the CCNA MapReducer with proposed data pre processing technique has less MAE value than the other techniques.

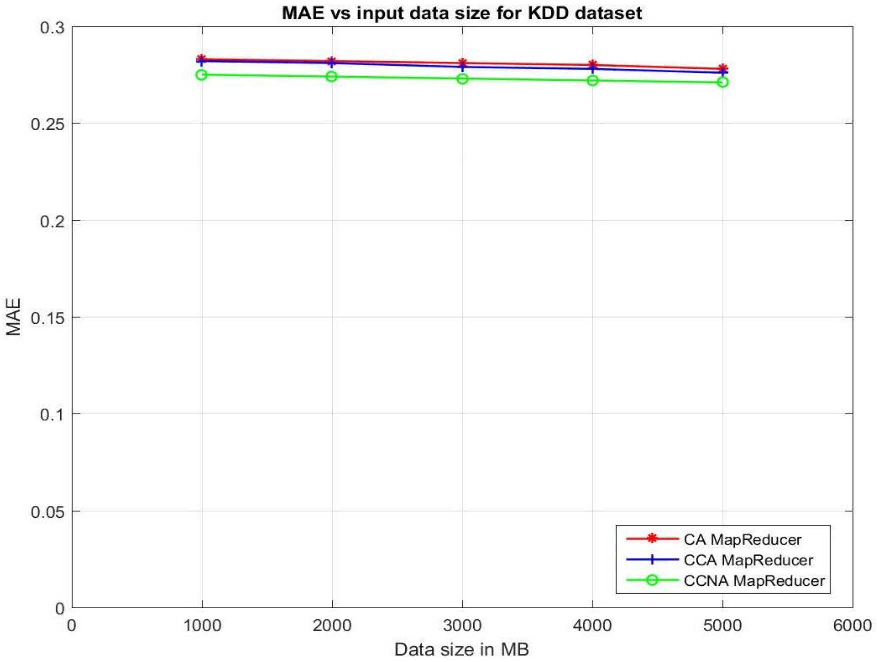

Figure 3.5 Mean Absolute Error for KDD Dataset

From the figure 3.6 and table 3.6 shows that MAE value against data size. The proposed CCNA MapReducer method is authenticated, that shows 0.271 MAE value but the existing CA MapReducer method shows 0.279 MAE value and CCA MapReducer method shows 0.276 MAE value at the point of 5000 mb data size. At the point of 2000 mb size the proposed CCNA MapReducer method shows 0.274 MAE value but the existing CA MapReducer method shows 0.282 MAE value and CCA MapReducer method shows 0.281 MAE value for lipid profile dataset. From this it is found that the CCNA MapReducer with proposed data pre processing technique has less MAE value than the other techniques.

Table 3.6 Mean Absolute Error for Lipid Profile Dataset

Input data size in MB	RMSE		
	CA MapReducer	CCA MapReducer	CCNA MapReducer
1000	0.279	0.278	0.274
2000	0.278	0.276	0.273
3000	0.277	0.275	0.272
4000	0.276	0.274	0.271
5000	0.275	0.273	0.270

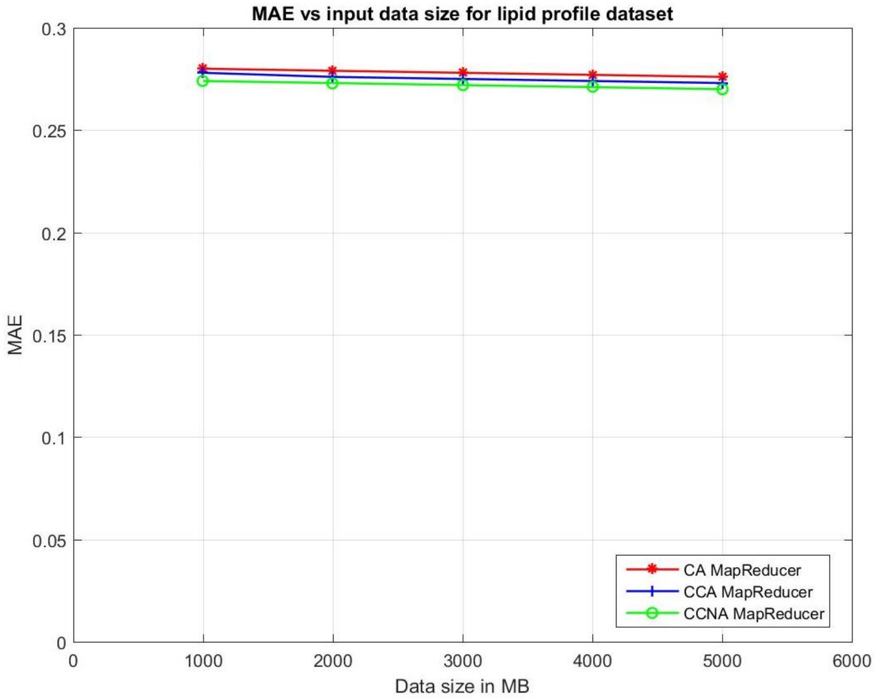

Figure 3.6 Mean Absolute Error for Lipid Profile Dataset

From the experimental results, it is proved that the proposed CCNA MapReducer performs better than the existing CA MapReducer and CCA MapReducer.

4. Conclusion

In this work, the noisy data, irrelevant data and missing data in big data are handled which decreases the process of big data. The missing data values are imputed by using K nearest neighbor algorithm which imputes the missing value in columnar wise with less computational cost and less time consumption. Additionally the noise in dataset is removed in row wise by using K nearest neighbor algorithm. Thus the proposed column wise missing value imputation and row wise noise removal works more effectively based on the content consideration, computation and network data aware than the existing methods. Further, there is considerable scope for further improvement.

References

[1] Karthick, N., and X. Agnes Kalarani. "An Improved Method for Handling and Extracting useful Information from Big Data", Indian Journal of Science and Technology, (2015).

[2] Kumar, K. S., Ramkumar, S., Shankar, K., & Ilayaraja, M. “Improving Mapreduce Process By Introducing Aggregator Repartition Data for Big Data Analytics”. reading, 10, 11.

[3] Karthick, N., & Kalarani, X. A,. “An Effective Filtering on Big Data by Finding Relevant Features to Extract Useful Information”, International Journal of Applied Engineering Research, 11(6), (2016), pp. 3805-3810.

[4] Nirmal, V. J., & Amalarethinam, D. G, “Parallel Implementation of Big Data Pre-Processing Algorithms for Sentiment Analysis of Social Networking Data”, International Journal of Fuzzy Mathematical Archive, 6(2), (2015) , pp.149-159.

[5] García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. “Big data preprocessing: methods and prospects. Big Data Analytics”, 1(1), 9, (2016).

[6] Patro, S., & Sahu, K. K, “Normalization: A Preprocessing Stage”, arXiv preprint arXiv:1503.06462, (2015).

[7] Malarvizhi, M. R., & Thanamani, A. S,” K-nearest neighbor in missing data imputation”, International Journal of Engineering Research and Development, 5(1),(2012),pp. 5-7.

[8] Batista, G. E., & Monard, M. C,” A Study of K-Nearest Neighbour as an Imputation Method”, HIS, 87(2002), 48, pp.251-260.

[9] Chen, L., Gao, Y., Chen, G., & Zhang, H. Metric,” All-k-Nearest-Neighbor Search”, IEEE Transactions on Knowledge and Data Engineering, 28(1), (2016) , pp.98-112.

[10] Das, T. K., & Kumar, P. M, “Big Data analytics: A framework for unstructured data analysis”, International Journal of Engineering Science & Technology, 5(1),(2013), pp.153.

[11] Grzymala-Busse, J. W., & Hu, M, “A comparison of several approaches to missing attribute values in data mining”, In International Conference on Rough Sets and Current Trends in Computing, (2000), pp.378-385.

[12] Islam, M. J., Wu, Q. J., Ahmadi, M., & Sid-Ahmed, M. A, “ Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers”, In Convergence Information Technology, International Conference on IEEE, (2007) , pp.1541-1546.

[13] Jadhav, S. D., & Channe, H. P, “Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques”, In International Journal of Science and Research (IJSR), 5(1), (2016), pp. 1842-1845.

[14] Wu, X., Zhu, X., Wu, G. Q., & Ding, W, “ Data mining with big data”, IEEE transactions on knowledge and data engineering, 26(1),(2014), pp. 97-107.

[15] Karthick, N., & Kalarani, X. A, “ An improved method for handling and extracting useful information from Big data”, Indian Journal of Science and Technology, 8(33), (2015), pp.1-7.

[16] Chen, B. W., Rho, S., Yang, L. T., & Gu, Y,” Privacy-preserved big data analysis based on asymmetric imputation kernels and multiside similarities”, Future Generation Computer Systems, 78, (2018), pp.859-866.

[17] Aktaş, M. S., Kaplan, S., Abacı, H., Kalipsiz, O., Ketenci, U., & Turgut, U. O,” Data Imputation Methods for Missing Values in the Context of Clustering”, In Big Data and Knowledge Sharing in Virtual Organizations, (2019), pp. 240-274, IGI Global.