ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 7, Issue 6, June 2018

Data Cleaning of Medical Datasets Using Data Mining Techniques

Usha T

Assistant Professor, Department of Computer Science & Engineering,
Sambhram Institute of Technology, Bangalore, India

Abstract: Data cleaning is a process that detects and removes the errors and inconsistencies in the data in order to improve the quality of the data. To have a high data quality, data quality problems has to be solved. Data quality problems exist in single and multiple source systems. A single source problem refers to the errors, inconsistencies, missing values, uniqueness violation, duplicated records and referential integrity violations. Multiple source problems are structural conflicts, naming conflicts, inconsistent timing and aggregating. In this paper, data quality problems such as duplication, missing values and attribute correction are solved by implementing different algorithm using data mining techniques.

Keywords: Data cleaning, Duplication, Missing data, Attribute correction, Levenshtein distance.

I. Introduction

Data mining is a process that is used to extract hidden patterns from the huge data set. The steps involved in data mining are Selection, Pre-processing, Transformation, Data Mining and Interpretation/Evaluation. One of the major steps is data pre-processing. Data cleaning [1] (data cleansing or data scrubbing) is one of the sub steps in data pre-processing. The intention of data cleaning process is to detect and remove errors and inconsistencies from the data in order to improve the quality of the data.

Data Cleaning [2] is a process used to determine inaccurate, incomplete, inconsistent and unreasonable data, then improving the quality of the data by correcting the detected errors. The sources of errors are lexical errors, syntactical errors, irregularities, duplicates and data entry anomalies.

Data plays a fundamental role in every software system. In particular, information system and decision support system depends on it more deeply. In today's environment, there is a need for more correct information for a better decision making [3]. Data quality is a crucial factor in data warehouse creation and during data integration. Medical data mining has got a great potential for exploring the hidden patterns in the data sets of medical domain. This pattern has to be utilized for clinical diagnosis. But there exists a data quality issues such as duplicated records, missing values, inconsistencies, referential integrity violations, uniqueness violations and errors that need to be handled in order to have high data quality.

Our approach focuses on handling data quality problems using different data mining techniques. Here, we present a brief overview on the area of data quality, categorization of data quality problems and different data cleaning algorithms designed using data mining approaches. We also present the results obtained using different algorithms.

II. Background

A. Problems With Data

Duplicated data: Duplication of a data occurs in two ways – duplication due to repeated records with some values different or different identification of the same real world entity.
Missing data: Missing values occurs in two ways – data are expected but are absent or data are inapplicable in the real world.
Erroneous data: This is due to incorrect value recorded for a real world value.

B. Data Quality

Data quality [4] specifies a state of completeness, consistency, validity, accuracy and timeliness that makes data appropriate for a specific use. Poor data quality leads to loss of money and inaccurate decision.

The following are the characteristics of data quality:

- **Completeness:**It is the characteristic of having filled all the required values for the data fields.
- **Validity:** It is a measure of degree of conformance of data values to its domain and rules. This includes Domain values, ranges, reasonability test, primary key uniqueness, referential integrity.
- **Accuracy:**It is a measure of the degree to which data agrees with data contained in an original source.
- **Precision:**The domain values should have correct precisions as per specifications.
- **Non-duplication:**It is the degree to which there is a one-to-one correlation between records and the real-world object or events being represented.
- **Derivation Integrity:**It is the correctness with which two or more pieces of data are combined to create new data.
- **Accessibility:**It is the characteristic of being able to access data as needed.
- **Timeliness:**It is the availability of data to support a given process as the time changes.

The hierarchy of data quality is as shown in Fig.1.

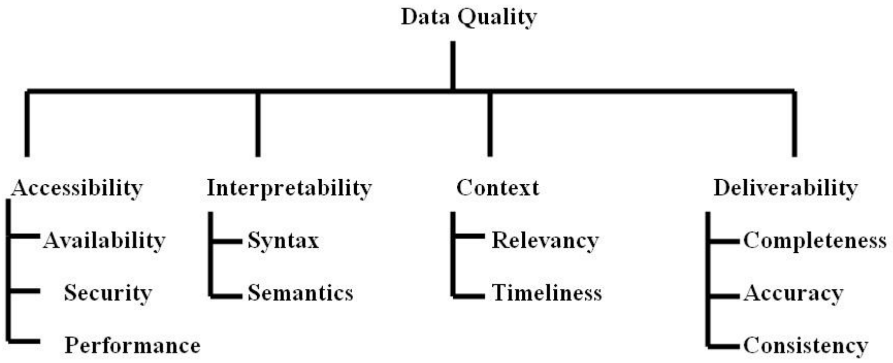

Fig. 1.Data quality hierarchy.

C. Data Quality Issues

Data quality issues [5] are classified as single source and multi-source problems. Further single source and multi-source problems are classified as schema level and instance level problems. Schema-level problems [6] are addressed at schema level by improving the schema design, schema translation and integration. Instance-level problem [7] refers to the errors and inconsistencies in the data set that are not visible at schema level. These problems are the primary focus in data cleaning. Fig. 2 shows the categorization of data quality problems.

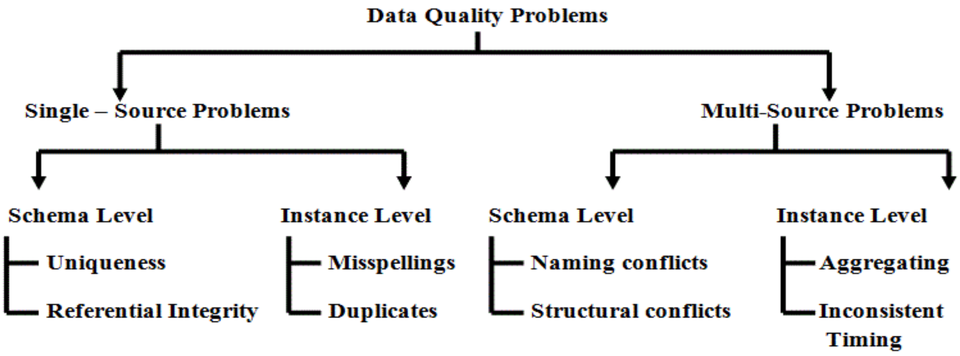

Fig. 2. Categorization of data quality problems in data sources.

As shown in the Fig. 2, single source problems refer to the errors, inconsistencies, missing values, uniqueness violation, duplicated records and referential integrity violations. Multiple source problems are structural conflicts, naming conflicts, inconsistent timing and aggregating.

III. Methodology

Here we focus on methods used for duplicate detection, missing value computation and attribute correction.

A. Duplicate detection

Duplicates [8] are detected using domain dependent algorithm [9]. In this algorithm domain knowledge is required.

DOI 10.17148/IJARCCE.2018.7612

The algorithm consists of following steps:

- Determine the uniqueness of each attribute in the dataset. Uniqueness is a numerical value which returns the number of different data of an attribute.
- Start sorting with the unique attribute.
- If the unique attributes are repeated then mark as duplicated and delete the record.

B. Missing value computation

When applying data mining approaches to the real-world data, learning from the incomplete data is a difficult situation. Trying to complete these missing values is a solution. There exists a numerous method to deal with missing values.

The following approaches [10] are used to find the missing attribute values:

- Use most common attribute value: The value of the attribute that occurs most frequently is selected to be the value for all the unknown values of the attribute.
- Use most common attribute value for all the samples belonging to the same class as the given tuple: The value of the attribute which occur the most common within the class, is selected to be the value for all the unknown values of the attribute.
- Use the attribute mean to fill in missing values: Mean of all the attributes can be used to fill in missing values.
- Use attribute mean for all the samples belonging to the same class as the given tuple:
  Calculate the mean within the class. The mean value can be used to fill in within the class.
- Fill in the missing value manually: In general, this approach is time consuming and may not be feasible given large data sets with many missing values.
- Method of Ignoring: This is usually done when the class label is missing.
- Use a global constant to fill in the missing value: Replace all the missing values by some global constant such as a label like “Unknown” or “∞”.
- Use the most probable value to fill in the missing value: This may be determined by using decision tree algorithm or linear regression methods.

Three different methods of dealing with missing values are studied i.e., Mean method, Linear regression and Lagrange interpolation method.

a. *Mean method*: Mean of all the attributes can be used to fill in missing values.

Algorithm: Mean(C [0…..n-1])
// Computes the missing value for the attribute cholesterol by taking the mean of cholesterol values.
// Input: Database D, of medical datasets; An array C [0…n-1] of cholesterol.
// Output: The value of Cholesterol.
Method:

```
1.    sum ← 0; count ← 0
2.    for i← 0 to n-1 do
3.          sum ←sum + C[i]
4.          count++
5.    mean ← sum / count
6.    return mean
```

There are various methods for computing missing values. One among them is the Mean value method. In step 2-4, the algorithm finds the sum of all the cholesterol values and counts the number of records. Mean value calculated in step 5, is the value of cholesterol. Since Mean method is simple, we go for a most probable method called linear regression and Lagrange interpolation method.

b. *Linear Regression*:

A regression [11] is a statistical analysis for assessing the association between two variables. Regression is used to find the relationship between two variables.In the regression method, a regression model is fitted for each variable with the missing values.
Regression Equation y = a + b*x, where ‘b’=slope, ‘a’=Intercept.

$$\mathbf{a=}\ \frac{(\Sigma y)(\Sigma x^2)-(\Sigma x)(\Sigma xy)}{n(\Sigma x^2)-(\Sigma x)^2}$$

$$\mathbf{b=}\ \frac{n(\Sigma xy)-(\Sigma x)(\Sigma y)}{n(\Sigma x^2)-(\Sigma x)^2}$$

Where x and y are variables, b is the slope of the regression line, a is the intercept point of the regression line and the y axis, N is the number of values or elements, X is First Score, Y is Second Score, ΣXY is Sum of the product of first and Second Scores, ΣX is Sum of First Scores, ΣY is Sum of Second Scores, ΣX^2 is Sum of square First Scores.

Pseudo-code for the regression algorithm is as follows:
Algorithm: LinearRegression (B [0….n-1], C [0….n-1])
// Computes the missing value for the attribute cholesterol by taking the values of the attributes blood pressure and cholesterol.
// Input: Database D, of medical datasets; an array B [0…n-1] of blood pressure; an array C [0…n-1] of cholesterol.
// Output: The value of Cholesterol.
Method:

```
1.    sumX ← 0; sumY ← 0; sumXY ← 0; sumX$^2$ ← 0; count ← 0
2.    for i ← 0 to n-1 do
3.    sumX ← sumX + B[i]
4.    sumY ← sumY + C[i]
5.    xy ← B[i] * C[i]
6.    sumXY ← sumXY + xy
7.          x$^2$ ← B[i] * B[i]
8.          sumX$^2$ ← sumX$^2$ + x$^2$
9.          count++
10.    b ← ((count * sumXY) – (sumX * sumY)) / ((count * sumX$^2$) – (sumX)$^2$)
11.    a ← ((sumY – (b * sumX)) / count
12.    y ← a + b * x
13.    return y
```

In this algorithm step 2-9, computes the value for ΣX, ΣY, ΣXY, ΣX^2 and count the number of records. Step 10-11, computes the slope and intercept point of the regression line respectively using the values calculated in steps 2-9. The value y calculated in the step 12, is the value of cholesterol.

c. *Lagrange Interpolation*:

Interpolation [12] is a process of finding the unknown values from a known value. By taking values (xi, yi), where i= 0, 1.......n of any function Y = f(x), the process of estimating the values of 'y', for any intermediate value of 'x' is called interpolation.

Lagrange interpolation equation:

$$P(x) = \sum_{j=1}^{n} Pj\ (x)$$

$$Pj(x) = yj \prod_{\substack{k=1 \\ k\neq j}}^{n} \frac{x - xk}{xj - xk}$$

Pseudo-code for the Lagrange interpolation algorithm is as follows:
Algorithm: Lagrange Interpolation (B [0….n-1], C [0….n-1])
// Computes the missing value for the attribute cholesterol by taking the values of the attributes blood pressure and cholesterol.
// Input: Database D, of medical datasets; an array B [0…n-1] of blood pressure; an array C [0…n-1] of cholesterol.

// Output: The value of Cholesterol.

Method:

```
1.  sum ← 0; lag ← 1
2.  for i← 0 to n-1 do
3.      for j ← 0 to n-1 do
4.          if i ≠ j then
5.              lag ←(( x – B[i]) / (B[j] – B[i]) * lag
6.          sum ←sum + lag * C[j]
7.  return sum
```

Here step 2-6, computes the sum by using Lagrange interpolation formula. The value of sum calculated in the step 6 is the value of cholesterol.

The missing values for the attribute cholesterol are filled by taking the average of values computed by using Mean, Linear regression and Lagrange interpolation method.

C. Attribute correction

In attribute correction [13] we used different methods of data mining they are Clustering techniques for Context-independent correction, Associations rule for Context dependent correction and Hybrid error correction technique.

a. Clustering technique – Context-independent correction

In Context-independent, the attribute values are examined and corrected without regard to the other attributes in the data set. The main idea of this algorithm is based on the observation that in most of the data sets there are certain values having higher number of occurrences within the data sets and also very large number of attributes with a very low number of occurrences. Hence the most representative values are the reference data. The attribute values with low number of occurrences are considered as noise or misspelled data.

Levenshtein distance (LD) [14] is a measure of similarity between the two strings, which is referred as the source string (s) and the target string (t). The distance is the number of deletion, insertions, or substitutions required to transform the source string into the target string.

For instance,
• If s is “angina" and t is "angina", then LD(s,t) = 0, since no transformations are required. The two strings are already identical.
• If s is "anvina" and t is "angina", then LD(s, t) = 1, since one substitution (change "v" to "g") is required to transform source into target string.

The following is the modified Levenshtein distance

$$L\hat{e}v(s1,s2)=\frac{1}{2}.(\frac{Lev(s1,s2)}{\| s1\|}+\frac{Lev(s1,s2)}{\| s2\|})$$

The modified Levenshtein distance for strings is interpreted as an average fraction of one string that has to be modified to be transformed into others. For example, the LD between “Asymptomatic” and “Asyoptonatic” is 2 and modified Levenshtein distance is 0.25.

The algorithm uses two parameters:
1. Distance Threshold (distThresh): is the minimum distance between the two values allowing them to be marked as similar and related.
2. Occurrence Relation (occRel): is used to determine whether both values compared belong to the reference data set.

Pseudo-code for the Context-independent algorithm is as follows:
Algorithm: Context Independent (C [0….n-1])
//Detects and correct the errors of the attribute chest pain type with the help of reference dataset.
// Input: Database D, of medical datasets; An array C [0…n-1] of chest pain type; Distance threshold, distThresh;

```
//Occurrence relation, occRel.
// Output: Corrected values of chest pain type.

Method:
1.      for i $\leftarrow$ 0 to n-1 do
2.          value $\leftarrow$ count the occurrence of each C[i]
3.          if value >occRel then
4.      add to reference dataset R[i]
5.      for each chest pain type $C_j$ in database do
6.          distance $\leftarrow$LevenshteinDistance(R[j],C[j])
7.      if distance $\leq$ distThresh then
8.                  C[j] $\leftarrow$ R[j]
9.      return C[j]
```

In this algorithm Step 1-2, finds the number of occurrences of each chest paint type attribute. In step 3-4, the value of a attribute is taken as a reference data set if the value of occurrence computed is greater than the given occurrence relation threshold. In step 5-8, Levenshtein distance is computed between reference dataset and each of the chest pain type attribute values present in the database. Step 8 replaces the error value with reference data set value by comparing whether the computed Levenshtein distance is less than or equal to given distance threshold in step 7.

b. Association Rule Methodology – Context-dependent correction

In Context-dependent [15], the attributes are examined and corrected by taking the consideration of values of other attributes within a given record. Here we use association rule to discover valid rules for the datasets. Apriori algorithm is used to generate frequent item set.

The algorithm uses two parameters such as Minimum support (minSup) and Distance Threshold (disThresh). Levenshtein distance and modified Levenshtein distance are used which is same as discussed in the previous algorithm.

Pseudo-code for the Context-dependent algorithm is as follows:

```
Algorithm Context Dependent (A [0….n-1], B [0….n-1], D [0….n-1], H [0….n-1], C [0….n-1])
// Detects and correct the errors by considering values of other attributes like age, blood pressure, cholesterol, heart
   beat.
// Input: Database D, of medical datasets; An array A [0…n-1] of age;
// An array B [0…n-1] of blood pressure; An array D [0…n-1] of cholesterol;
// An array H [0…n-1] of heart beat rate; An array C [0…n-1] of chest pain type;
// Minimum support, minSup; Distance threshold, distThresh.
// Output: Corrected values of chest pain type.

Method:
    $L_1 \leftarrow$ {frequent items}
    for (k=0;$L_k \neq \phi$; k++) do
        $C_{k+1} \leftarrow$ Candidates generated from $L_k$
    for each transaction t in D do
    $c_t \leftarrow$ subset ($C_k$, t)
                for each candidate $c \in C_t$
    c.count++
        $L_k \leftarrow \{c \in C_k \mid c.\ count \geq minSup\}$
    return $\cup_k L_k$
     for each frequent itemset i do
         for each subset s of i do
             conf $\leftarrow$ support(i) / support(i - s)
             if conf $\geq$ minConf then
                 Output the rule (i-s) $\rightarrow$ s
     for each chest pain type $C_j$ in database do
         distance $\leftarrow$LevenshteinDistance(S[j],C[j])
         if distance $\leq$ distThresh then
             C[j] $\leftarrow$ S[j]
     return C[j]
```

In this algorithm step 1 finds the frequent 1-itemsets L_1. Steps 2-8, is used to generate candidates C_{k+1} in order to find L_k. Steps 10-14 generates the association rule for the sets generated in the steps 1-8. In step 14, the rules generated may have one, two or three predecessor and one successor. In steps 15-18, the algorithm checks whether the predecessor is same and successor is different then it computes the Levenshtein distance between the successor and chest pain type attribute values present in the database(step 16). If the Levenshtein distance is lesser than or equal to distance threshold, the values are corrected (step 18).

c. *Hybrid Error Correction Technique (HECT)*

A new algorithm HECT has been proposed, which combines the concept of both context-independent and context-dependent for data standardization and correction. Using this approach we can correct the errors more accurately compared to other two algorithms.
The algorithm uses parameters such as Occurrence Relation (occRel), Minimum support (minSup) and Distance Threshold (disThresh). Levenshtein distance and modified Levenshtein distance are used which is same as discussed in the previous algorithm.

Pseudo-code for the HECT is as follows:
AlgorithmHECT (A [0….n-1], B [0….n-1], D [0….n-1], H [0….n-1], C [0….n-1])
// Detects and correct the errors by considering values of other attributes like age, blood pressure, cholesterol, heart beat and also with the help of reference dataset.
// Input: Database D, of medical datasets; An array A [0…n-1] of age;
// An array B [0…n-1] of blood pressure; An array D [0…n-1] of cholesterol;
// An array H [0…n-1] of heart beat rate; An array C [0…n-1] of chest pain type;
// Minimum support, minSup; Distance threshold, distThresh; Occurrence relation, occRel.
// Output: Corrected values of chest pain type.

Method:

```
1.      for i← 0 to n-1 do
2.          value ← count the occurrence of each C[i]
3.          if value >occRel then
4.      add to reference dataset R[i]
5.      $L_1$ ← {frequent items}
6.      for (k=0;$L_k \neq \phi$; k++) do
7.           $C_{k+1}$ ← Candidates generated from $L_k$
8.      for each transaction t in D do
9.      $c_t$← subset ($C_k$, t)
10.                 for each candidate $c \in C_t$
11.     c.count++
12.          $L_k$← {$c \in C_k$ | c. count $\geq$ minSup}
13.      return $\cup_k L_k$
14.      for each frequent itemset i do
15.          for each subset s of i do
16.                conf ← support(i) / support(i - s)
17.                if conf $\geq$ minConf then
18.                     Output the rule (i-s) → s
19.     for each chest pain type $S_j$ do
20.          distance ←LevenshteinDistance(S[j], R[j])
21.          if distance $\leq$ distThresh then
22.                 S[j] ← R[j]
23.      return S[j]
```

In this algorithm Step 1-2, finds the number of occurrences of each chest paint type attribute. In step 3-4, the value of a attribute is taken as a reference data set if the value of occurrence computed is greater than the given occurrence relation threshold. Step 5 finds the frequent 1-itemsets L_1. Steps 6-13, is used to generate candidates C_{k+1} in order to find L_k. Steps 14-18 generates the association rule for the sets generated in the steps 5-13. In step 18, the rules generated may have one, two or three predecessor and one successor. In the steps 19-23, if the Levenshtein distance computed is lesser than or equal to the given distance threshold, the values are corrected as shown in the step 22.

IV. RESULTS

The algorithm was tested using a sample Cardiology dataset which is from Hungarian data. The attribute chest pain type (CP) is the source for data cleaning. Table I shows the example transformation rules discovered during the execution of the algorithm.

Table I
Example Transformation Rules

Original value	**Correct value**
Asmytomatic	Asymptomatic
Asmythmatics	Asymptomatic
Assymtomatics	Asymptomatic
Asympotmatic	Asymptomatic
Asymptomac	Asymptomatic

Fig. 3 show the graph for the techniques used in attribute corrections. Using context-independent concept, 54% of records are corrected successfully and 15% of records are not altered. The number of records corrected is increased with the use of context-dependent concept i.e., 62% of records are corrected successfully and only 7% of records are not altered. In Hybrid error correction technique (HECT), all the records are corrected successfully.

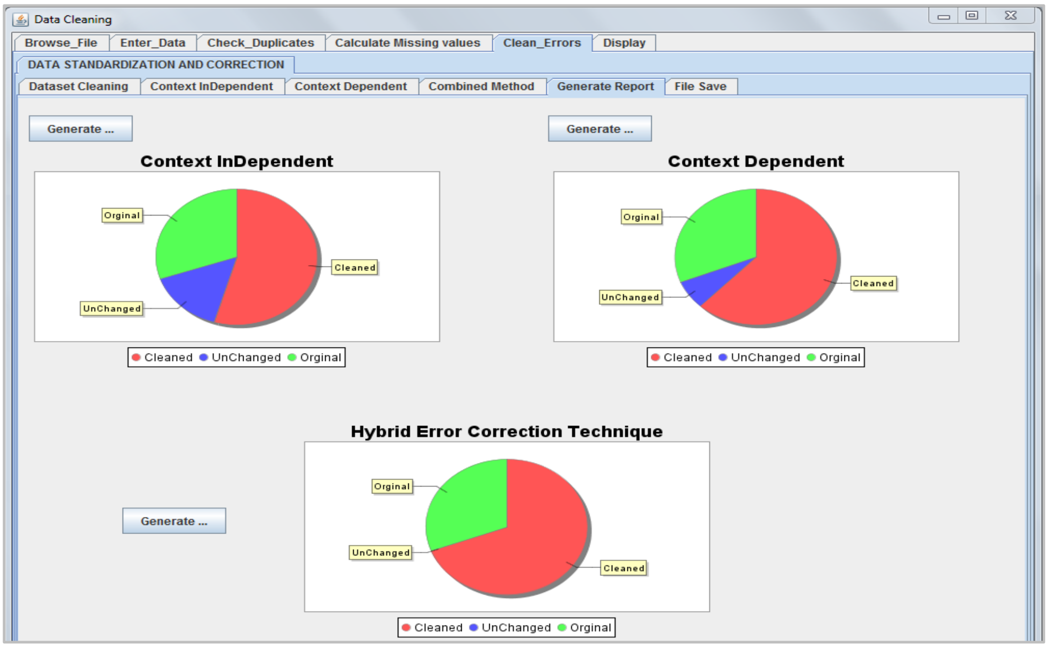

Fig. 3. Graph for comparing data cleaning algorithms.

The following Table II shows the percentage of records corrected, uncorrected and existentially valid records by using context-independent, context dependent and hybrid error correction techniques.

Methods	Percentage of records		
	Cleaned	Not cleaned	Originally Correct
Context-independent	54 %	15 %	31%
Context dependent	62%	7 %	31%
Hybrid error correction technique	69%	_	31%

Table II: Comparison of different data cleaning algorithms.

DOI 10.17148/IJARCCE.2018.7612

Fig. 4 show the graphical representation of the methods compared in the above Table II. From this graph we can infer that hybrid error correction technique yields the best results.

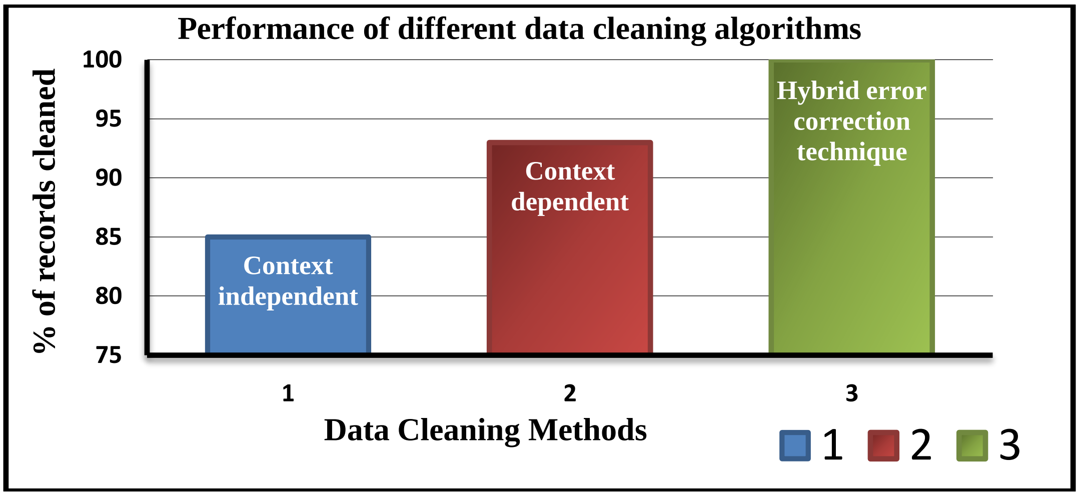

Fig.4. Performance analysis of data cleaning algorithm

V. Conclusion

Data or the information that are used in the medical field is one of the most important assets. High data quality is one of the most important requirements for taking successful decisions over the medical datasets. In order to provide high data quality for these datasets we have implemented different data mining techniques. Data quality issues such as duplicates, missing values and errors are solved using these different techniques making the data efficient and accurate. From the above result it is observed that the result of hybrid error correction technique is best than the context dependent attribute correction and context independent attribute correction.

In this work we have implemented the algorithms to find duplicates, missing values and errors using single source data. Further as a future work, multi-source problems such as naming conflicts and structural conflicts can be solved by collecting the data from different sources.

References

[1] KDnuggets Polls. “Data Preparation Part in Data Mining Projects”, Sep30-Oct-12, 2003. http://www.kdnuggets.com/polls/2003/data_preparation.htm.
[2] Lee, M.L.; Lu, H.; Ling, T.W.; Ko, Y.T.: Cleansing Data for Mining and Warehousing. Proc. 10th Intl. Conf. Database and Expert Systems Applications (DEXA), 1999.
[3] Paul Jermyn, Maurice Dixon and Brian J Read, “Preparing clean views of data for data mining”.
[4] Sweety Patel” Requirement to cleanse DATA and why is data cleansing in Business Application?” International Journal of Engineering Research and Applications, Vol. 2, Issue 3, May-Jun 2012.
[5] Erhard Rahm, Hong Hai Do. “Data Cleaning: Problems and Current Approaches”. IEEE Data Engineering Bulletin, 2000, 23(4):3-13.
[6] Wang Y.R. ;Madnick S.E, “The inter-database instance identification problem in integrating autonomous systems” Proceedings of the Fifth International Conference on Data Engineering, IEEE Computer Society, Silver Spring 1999, February 6–10, 1999, Los Angeles, California, USA, p. 46–55.
[7] Adam Widera, Michal Widera, Daniel Feige “ Data Cleaning on Medical Data Sets“ Journal of Medical Informatics and Technologies,Vol. 8,ISSN 1642-6037, Jun e 2008.
[8] Mauricio Hernandez, Salvatore Stolfo, “Real World Data Is Dirty: Data Cleansing and The Merge/Purge Problem”, Journal of Data Mining and Knowledge Discovery, 1(2), 1998.
[9] Dr. PayalPahwa, Rashmi Chhabra,”Domain Dependent and Independent Data Cleansing Techniques“, International Journal of Computer Science and Telecommunications, Vol. 2, Issue 3, September 2011.
[10] Jerzy, W.Grzymala-Bussel and Ming” Comparison of Several Approaches to Missing Attribute Values in Data Mining Techniques” Journal of Computer Science ISSN 1549-363 Science Publications.
[11] Z. Mahesh Kumar , R. Manjula “Regression model approach to predict missing values“ International Journal of Computer Science & Engineering Technology , ISSN : 2229-3345 Vol. 3 No. 4 April 2012.
[12] L.Sunitha, Dr M.BalRaju , J.Sasikiran “Estimation of Missing Values Using Lagrange Interpolation Technique “ International Journal of Advanced Research in Computer Engineering & Technology, Volume 2, Issue 4, April 2013.
[13] R. Kavitha Kumar and Dr. RM. Chadrasekaran “Attribute Correction – Data Cleaning using Association rule And Clustering Methods” International Journal of Data Mining & Knowledge Management Process ,Vol.1, No.2, March 2011.
[14] W. Cohen, P. Ravi Kumar, S. Fienberg “A Comparison of String Metrics for Name-matching Tasks” in Proceedings of the IJCAI-2003.
[15] Lukasz Ciszak, “Application of clustering and Association Methods in data cleaning”, 978-83-60810-14-9/08, 2008 IEEE.

DOI 10.17148/IJARCCE.2018.7612

BIOGRAPHY

USHA T
Assistant Professor, Department of CSE, Sambhram Institute of Technology, Bangalore, India.

DOI 10.17148/IJARCCE.2018.7612