BASMA AL-BRGE

A HYBRID METHOD FOR MISSING VALUE IMPUTATION

BASMA AL-BRGE

31 JANUARY 2019

ATILIM UNIVERSITY

A HYBRID METHOD FOR MISSING VALUE IMPUTATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

BASMA AL-BRGE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INFORMATION TECHNOLOGY

31 JANUARY 2019

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

Prof. Dr. Ali KARA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Information Technology, Atilim University**.

Assoc. Prof. Dr. Korhan Levent ERTÜRK
Head of Department

This is to certify that we have read the thesis **A HYBRID METHOD FOR MISSING VALUE IMPUTATION** submitted by **BASMA AL-BRGE** and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Murat KOYUNCU
Supervisor

Examining Committee Members:

Assoc. Prof. Dr. Tolga PUSATLI
Mathematics, Çankaya University

Assoc. Prof. Dr. Murat KOYUNCU
Information Systems Eng., Atilim University

Asst. Prof. Dr. Çiğdem TURHAN
Software Department, Atilim University

Date: 31 January 2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Basma Al-Brge

Signature :

ABSTRACT

A HYBRID METHOD FOR MISSING VALUE IMPUTATION

Al-Brge, Basma

M.Sc, Information Technology

Assoc. Prof. Dr. Murat Koyuncu

January 2019, 63 pages

Missing data arises in almost all serious statistical analyses. Statistical analyses have a variety of methods to handle missing data, including some relatively simple approaches that can often yield reasonable results such as the random imputation approach. The missing data imputation process must be modeled in order to perform imputations correctly. Using datasets in empirical applications is very common to perform some tasks; however, missing values in datasets should be extracted from the datasets or should be estimated before they are used for processing to produce correct association rules or clustering in the preprocessing stage of data mining and processing. In this thesis, a hybrid approach is used that combines K-Nearest Neighbor (KNN) with Singular Value Decomposition (SVD) algorithm to improve the data imputation and produce data with high correlation with original missing values. The test results of the proposed hybrid method are compared with the results of several alternative methods for different rate of missing values and the results of the proposed method yields better performance than the others. The results are also compared with the reported results in the literature to give an idea about its performance.

Keywords: Hybrid approach, Missing values, K-nearest Neighbour, Singular Value Decomposition.

ÖZ

KAYIP VERİLERİN TAMAMLANMASI İÇİN BİR HİBRİT MODEL

AL-Brge, Basma

Yüksek Lisans, Bilgi Teknolojileri

Doç. Dr. Murat Koyuncu

Ocak 2019, 63 Sayfa

Eksik veriler neredeyse tüm ciddi istatistiksel analizlerde ortaya çıkmaktadır. İstatistiksel analizler, eksik verileri işlemek için, rastgele değerlendirme yaklaşımı gibi genellikle makul sonuçlar verebilecek bazı basit yaklaşımlar da dahil olmak üzere çeşitli yöntemlere sahiptir. Eksik veri değerlendirme süreci, doğru tamamlamalar yapabilmek için modellenmelidir. Veri setlerini ampirik uygulamalarda kullanmak bazı görevleri gerçekleştirmek için çok yaygındır, ancak veri setlerindeki eksik değerler veri setlerinden çıkarılmalı ya da veri madenciliğinin ön işleme aşamasında tahmin edilmelidir. Bu tezde, veri algılamasını iyileştirmek ve orijinal eksik değerlerle yüksek korelasyonlu veri üretmek için K-En Yakın Komşu (KNN) ile Tekil Değer Ayrıştırma (SVD) algoritmasını birleştiren bir karma yaklaşım kullanılmaktadır. Önerilen hibrit yöntemin test sonuçları, farklı kayıp değerlerin oranı için çeşitli alternatif yöntemlerin sonuçlarıyla karşılaştırılmış ve önerilen yöntemin performansı diğerlerinden daha iyi çıkmıştır. Ayrıca sonuçlar, önerilen modelin performansı hakkında bir fikir vermesi amacıyla literatürdeki raporlanan diğer sonuçlarla da karşılaştırılmıştır.

Anahtar Kelimeler: Hibrit yaklaşım, Kayıp değerler, K-en yakın komşu, Tekil Değer Ayrışımı.

To the spirit of my father, the first to teach me

To my beloved mother, for her prayers to me

To my brother and sister, for care and support all the time

To my husband and love my life, Sabah for always showing how proud he is of me

To my children Zaid and Haider, the light of my life

To the people who paved our way of science and knowledge

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my supervisor Assoc. Prof. Dr. MURAT KOYUNCU for his support, guidance, encouragement, useful critiques for this research work and patience during my research. I owe great thanks to him. Also, I wish to express my thanks to all academic staff in the Department of Computer Engineering, Software Engineering and Information System Engineering at Atilim University for their effort and helps.

I wish thank my dear mother, brother Muhammed and my dear sister Farah for their love and encouragement, without whom I would never have enjoyed so many opportunities.

The last word goes for Zaid and Haider, my children , who have been the light of my life for the last three years and who has give me the extra streangth and motivation to get things done. This thesis is dedicated to them.

Last, but never least, I must thank my unbelievably supportive husband Sabah. He has demonstrated rare and amizing patience throughout my long working sessions over the last three years. Without his love and sacrifice, this thesis would not be possible.

TABLE OF CONTENTS

ABSTRACT iv
ÖZ v
DEDICATION vi
ACKNOWLEDGMENTS vii
TABLE OF CONTENTS viii
LIST OF TABLES x
LIST OF FIGURES xi
LIST OF SYMBOLS/ABBREVIATIONS xiii
CHAPTER 1
INTRODUCTION
1.1 Missing Value (MV) 1
1.2 Missing Data Mechanisms 2
1.2.1 Missing Completely At Random (MCAR) 2
1.2.2 Missing At Random (MAR) 2
1.2.3 Missing Not At Random (MNAR) 3
1.3 Research Topic 3
1.4 Related Studies 3
1.5 Contribution 5
1.6 Research Organization 5
CHAPTER 2 6
BACKGROUND AND LITRATURE REVIEW 6
2.1 Missing Data and Implication for a Research Finding 6
2.2 Missing Data Imputation Methods 7
2.2.1 Traditional Missing Data Imputations 7
2.2.1.1 Listwise Deletion 7
2.2.1.2 Pairwise Deletion 8
2.2.1.3 Single Imputation 9
2.2.2 Modern Missing Data Imputation 9

2.2.2.1 Multiple Imputation (MI) 10
2.2.2.2 Maximum Liklihood Imputation (ML) 11
2.3 Literature Review 12
2.4 K-Nearest Neighbor Algorithm (KNN) 14
2.5 Singular Value Decomposition (SVD) 16
CHAPTER 3 18
METHODOLOGY 18
3.1 Introduction 18
3.2 Research Framework 19
3.3 Proposed System for Filling Missing Values 20
3.4 Imputation Techniques 23
3.4.1 K-Nearest Neighbor (KNN) Algorithm 23
3.4.1.1 Case Study For (KNN) Imputation 25
3.4.2 Singular Value Decomposition (SVD) Algorithm 29
3.4.2.1 Case Study For (SVD) Imputation 30
CHAPTER 4 36
EXPEREMENT ENVIROMENT AND RESULTS 36
4.1 Introduction 36
4.2 Implementation Environment 36
4.3 Implementation of Proposed System 37
4.4 Case Study of Suggested Hybrid Method (KNNSVD) 38
4.5 Experiment Results 43
CHAPTER 5 54
CONCLUSION AND RECOMONDATIONS 54
5.1 Conclusions 54
5.2 Recommendations for Future Works 55
REFERENCES 56
APPENDICES
Appendix A- General Terms 61
Appendix B- Implimentation Screenshots 62

LIST OF TABLES

Table 2.1 Listwise Deleation Pattern 7
Table 2.2 Pairwise Deleation Pattern 9
Table 2.3 Summary of the study [22] 13
Table 2.4 Summary of the study [23] 14
Table 2.5 Summary of the study [24] 14
Table 3.1 Structure of Yeast Data 21
Table 3.2 Sample of Yeast Data 21
Table 3.3 Random Missing Values Generation 22
Table 3.4 Dataset with NaNs 22
Table 3.5 Splitting Process 23
Table 3.6 Testing Data (with missing values) for Example 1 25
Table 3.7 Training Data (without missing values) for Example 1 25
Table 3.8 Imputed Data for Example 1 26
Table 3.9 Testing Data for Example 2 (with missing values) 27
Table 3.10 Imputed Data for Example 2 29
Table 4.1 MSE Result of KNN and SVD 46
Table 4.2 MSE Result for Hybrid Techniques (KNNSVD, SVDKNN) 50
Table 4.3 MSE Results Comparison for the Tested Methods 52

LIST OF FIGURES

Figure 1.1 Missing Data Mechanisms 2
Figure 2.1 Missing Value Imputation Methods 6
Figure 2.2 KNN Mechanisms 15
Figure 3.1 Research Framework 19
Figure 3.2 Proposed System 20
Figure 4.1 Sample of Yeast Data 36
Figure 4.2 Generation of Missing Values 37
Figure 4.3 Imputation of Missing Values 37
Figure 4.4 Random Missing Values Generation 38
Figure 4.5 First Group with Threshold More than 20% 39
Figure 4.6 Second Group with Threshold Less than 20% 40
Figure 4.7 Real Value of NaNs in Figure 4.5 40
Figure 4.8 Real Value of NaNs in Figure 4.6 41
Figure 4.9 First Group after KNN Imputation 41
Figure 4.10 Second Group after SVD Imputation 42
Figure 4.11 Random Missing Values Generation 43
Figure 4.12 Replace NaN Values in Figure 4.11 with 1 44
Figure 4.13 Real Value of NaNs in Figure 4.12 45
Figure 4.14 KNN Imputation without Setting Threshold 45
Figure 4.15 SVD Imputation without Setting Threshold 46
Figure 4.16 First Group with Threshold More than 20% 48
Figure 4.17 Real Value of NaNs in Figure 4.16 48
Figure 4.18 SVD Imputation on the First Group 49
Figure 4.19 Second Group with Threshold Less than 20% 49
Figure 4.20 Real Value of NaNs in Figure 4.19 50
Figure 4.21 KNN Imputation on the Second Group 51
Figure 4.22 MSE Result Comparison between KNN, SVD, KNNSVD, SVDKNN 53
Figure 4.23 MSE Result Comparison between KNNSVD, SVDKNN 52
Figure 4.24 Error Comparison between KNNSVD and Previous Methods 53

LIST OF ABBREVIATIONS

AMMI	Additive Main Effect and Multiplicative Interaction
ANN	Artificial Neural Network
DFMI	Distribution Free Multiple Imputation
EM	Expectation Maximization
Fcm	Fuzzy c-mean cluster
Ga	Genetic Algorithm
GKNN	Gray K-Nearest Neighbor
KNN	K-Nearest Neighbor
LD	Listwise Deletion
MAE	Mean Absolute Error
MAIE	Mean absolute Imputation Error
MAR	Missing At Random
MCAR	Missing Completely at Random
MCMC	Markov Chain Monto Carlo
MI	Multiple Imputation
ML	Maximum Likelihood
MNAR	Missing Not At Random
MSE	Mean Squared Error
MSIE	Mean Square Imputation Error
MV	Missing Value
MVs	Missing Values
NRMSE	Normalize Root Mean Squared Error
PD	Pairwise Deletion
PSM	Pattern Similarity Matching
REM	Regularize Expectation -Maximization
SVD	Singular Value Decomposition
SVR	Support Vector Regression
SVr	Support Vector Regression
SVM	Support Vector Machine
WKNN	Weighted Nearest Neighbor

CHAPTER 1

INTRODUCTION

1.1 Missing Value (MV)

Machine learning, data mining and various information systems all avoid having MV [1]. In recent years, a lot of research is being done particularly in dealing with datasets having missing values like deleting, usage of zero or means estimation method and ignoring. Nevertheless, the main disadvantage of such estimation methods is efficiency loss because of biases in the estimates made as well as overlooking incomplete observations. High-quality data is the source of achieving higher quality mining outcomes and results. Thus, it is necessary to estimate and assess the missing values for achieving better and efficient data quality. The main reasons and conditions that cause missing values to arise are absence of digital response in scientific experiments, sensor faults, faulty measurements, data transfer problems in digital systems or unwillingness of responding by the respondents to survey questions particularly in scientific research. As a result, some variables may end up with missing values and, therefore, they become unavailable for analysis. In cases where a missing value is not dealt with appropriately, the authenticity of research is compromised. Therefore, one needs to have better insight in dealing with the mechanism of missing data and by doing this make it easy to access the fundamental reason of the mechanism for missing data. Such mechanisms which define underlying or fundamental reasons of missing data were for the first time explained by Rubin in 1987 [2]. He classified the mechanisms of missing data into three categories, as shown in Figure 1.1, which are missing not at random (MNAR), missing at random (MAR), and missing completely at random (MCAR), and all of these are very significant as they provide assumptions of missing data methods. Here it is very crucial to bring into notice that the missing data mechanisms are not typical of the whole data; rather, the mechanisms are just assumptions that appeal to certain methods of analysis.

1.2 Missing Data Mechanisms

There can be several causes that can create missing data in real-world databases, therefore, it is crucial in recognizing any pattern in the missing data as it is an important part of conceiving methods that deal with the missing observations. In particular, the quality of results yielded from predictions of classification methods applied on the data relies heavily on the kind of missing data. Here we recap the conventionally accepted types of missing data [3].

Figure 1.1 Missing Data Mechanisms

1.2.1 Missing Completely At Random (MCAR): is the one and the only mechanism of missing data which can, in fact, be verified. It is the missing data when the probability of missing data within a variable is not related to the measured variable itself [4]. Simply, it is unsystematic missingness on a certain variable like in the case when a certain data of respondents is missing after their questionnaire gets lost in the mail.

1.2.2 Missing At Random (MAR): is the missing data whose probability of occurring is affiliated to a different measured variable in the given model rather than the variable with values missing. As an example, in case of a dataset of older people whose age values have some missing values, the missing values could be correlated with the age of the old people whose age data is missing [5].

1.2.3 Missing Not At Random (MNAR): is when missing values themselves are the cause for missingness rather than being dependent on the variables. To explain it further, it is like in case of missing data on values of age where only the individuals with low age values possess missing observations for the variable [6].

1.3 Research Topic

Numerous studies have been done to enhance and control the limitations that come with the standard simple imputation methods. These methods include K-Nearest Neighbour (KNN), Support Victor Regression (SVR), Singular Value Decomposition (SVD), Artificial Neural Network (ANN), Mean, and Median. They are not able to give reliable results. This lack of reliability will be the focus of this thesis by introducing a technique more effective with regards to error rate. We propose a new imputation method using KNN and SVD in a hybrid way. Here we intend to enhance the quality of the predictions made regarding the missing values while avoiding any influence on other data characteristics. Different studies have shown that different algorithms create variable success rates that rely on missing ratios. For this reason, we propose to put two algorithms together in a special methodology to get a brand-new hybrid technique. We evaluate the effectiveness of the proposed method by measurement the Mean Squared Error (MSE). For experimental evaluation, the Yeast data set consisting of 8 columns and 1484 row from the UCI Machine Learning Repository is used.

1.4 Related Studies

Grey K.Nearest Neighbor (GKNN) imputation method was suggested to iteratively impute missing value [7]. Normally GKNN chooses K closest neighbor for every missing value by calculating the distance between the training data and MV using the grey distance (the iteration lacking MVs). This technique yielded better results as compared to traditional distance at both managing with mix data and holding proximity relationship of two distances (categorical and numerical data). Moreover, the result of the trial suggested that the GKNN has greater effectiveness than the existing KNN methods of imputation.

In [8], a hybrid method Fuzzy C-mean cluster Support Vector Regression Genetic algorithm (FC-mean SVr Ga) was applied when imputing missing data. Here the MVs were estimated using the known and reliable machine learning technique, SVR, as well as Ga, via FC-mean. Based on similarity, the training data (iterations lacking MV) were clustered using fuzzy principles. One or more cluster centroids can contain each of the MVs within them as a member. The suggested method was also compared with models including the FC-mean SVr Ga. The result of the comparison shows that the FC-mean SVr Ga bring about a far better adequate and accurate ratio for feasible clustering.

In another study, four imputation algorithms were applied on 3 real datasets composed of data from sugar cane, beans and eucalyptus [9]. The algorithms were based on SVD, namely (Biplot imputation, Expectation Maximization (EM) + SVD, Gabriel Eigen imputation and distribution-free multiple imputations DFMI). The application of these algorithms on the datasets yielded different imputation percentages. The results were compared with results from the gold standard EM and Additive Main Multiplication (AMMI) using Normalized Root Mean Squared Error (NRMSE). The outcome of this comparison proved EM algorithm and SVD to be great alternatives to AMMI in addition to yielding comparable results to those acquired with the gold standard.

In [10], weighted K-nearest neighbor (wKNN) imputation was suggested to impute missing values with use of L distance for measuring the distance separating any two observations, x_i and x_j, presented as a row in data matrix with 5 %, 10% and 25% proportions of missing values. wKNN has been demonstrated to have a smaller imputational error as compared with other estimates made using nearest neighbor methods. wKNN is particularly useful when we have a large number of predictors. MSIE and MAIE are used to evaluate the proposed method.

1.5 Contribution

This thesis raises the question of missing data and proposes a predictive technique to tackle missing data problem. Here, we present a brand-new technique using KNN and SVD in a hybrid form. This study gives a fresh insight into the obstacles of classical missing data and offers a better new technique for enhancing the quality level of missing value prediction.

1.6 Research Organization

Chapter One: Here, general introduction concerning the missing value, missing data mechanisms, studies and topics relating the current research and contributions are presented.

Chapter Two: This chapter deals with the literature review, imputation with Singular Value Decomposition (SVD), imputation with k-Nearest Neighbor, and missing data handling methods.

Chapter Three: In this section the research framework is presented along with the suggested system for filling missing values.

Chapter Four: In this chapter we focus on the implementations of system and evaluation of the proposed method's performance.

Chapter Five: The last chapter states conclusions and mentions recommendations for further work.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

This chapter presents the background of the proposed work, missing data imputation techniques and related studies.

2.1 Missing Data and Implication for a Research Finding

There can be some difficult challenges faced while doing research particularly in case where there is involvement of participants. Some of these challenges are like sufficient representation and measurement of target constructs (e.g. mathematical knowledge) and collecting representative samples of populations of interest. Apart from these, missing data can create problem as well. Regardless of acquiring certain sample of participants and their willingness to participate, one cannot control their full commitment to the research as they might withdraw or not participate completely i.e. answer the questions incompletely or inappropriately. In such cases, the acquired data matrix results create missing values. Due to all these reasons, biases can exist in estimations of sample statistics when missing values are not dealt properly [2,11].

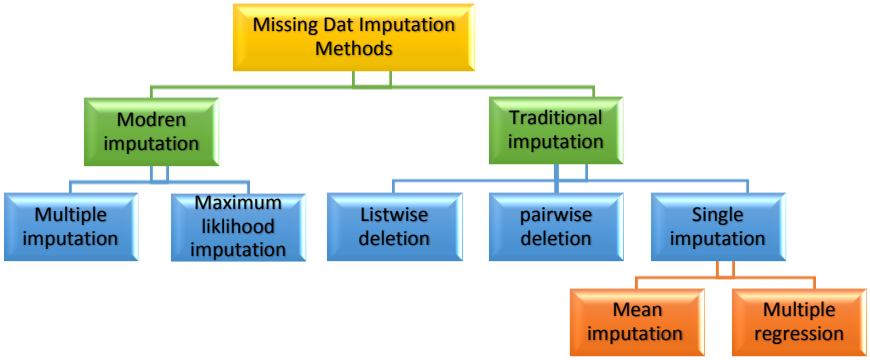


Figure 2.1 Missing Value Imputation Methods

2.2 Missing Data Imputation Methods

Missing data imputation methods can be classified as shown in figure 2.1 [12].

2.2.1 Traditional Missing Data Imputations

Methods like regression-based imputation, pairwise deletion, listwise deletion and mean imputation are the traditional methods of tackling missing data [12].

Table 2.1 Listwise Deletion Pattern

Subject	Age	Gender	Income
1	29	M	$55,000
2	50	NAN	$36,000
3	25	F	NAN

.

2.2.1.1 Listwise Deletion (LD)

It is deals with cases of variables with missing values in a given statistical model. LD also happens to be the default setting for missing treatment of data in numerous packages of software. LD may be a convenient method to use; however, its effectiveness at producing unbiased estimates is entirely dependent on the missing data mechanism. Moreover, LD results in reduced statistical power. Within the context of MCAR, LD provides a route to estimates which are unbiased especially when the missing data can be perceived as a simple random sample of the data matrix which is complete [13,14]. As suggested earlier under MCAR the probability of a chosen variable having a missing value is entirely independent of both unobserved and observed variables. This suggests that there is an equal probability for at least one missing variable to be expected on any given variable. In other words, we can expect at least one missing value to occur in any variable within the statistical model. This would be considered to be a random sample from a sampled case and would also be excluded in LD. Consequently, sample statistics resulting from the data of interest can be considered to have unbiased estimates under MCAR. Here

Table 2.1 shows the Listwise deletion pattern in which the researchers would delete subject 2 and 3.

2.2.1.2 Pairwise Deletion (PD)

Pairwise deletion or PD is a different approach to LD. PD uses all available data by just excluding mising values from calculations. For example, lets think three continuous variables X1, X2, X3 where all represent as a given proportion of missing data. The correlation between X1 and X2 under pairwise deletion, would be determined on those cases that possessed values for both variables X1 as well as X2 irrespective of whether those cases that possessed values for both variables X1 and X2 and also irrespective of whether those cases owned missing values on X3. Similarly, X1 and X3 correlation would be dependent on cases with values for X1 and X3 on both these variables irrespective of whether those cases had missing values for X2. This should be kept in mind, PD can happen in another sample size for the separate correlations as they are not depending on simply single sample. Generally, the MCAR data sample statistics are unbiased when based on PD. In contrast to LD, PD can give problem which are computational like covariance matrices, which are not of full rank [6,7,8]. Missing data treatment by LD as well as PD are ad hoc missing data treatments. These treatments are mostly applied by default while overlooking the missing data nature [9]. In case of data not being MCAR and researchers use LD or PD for the missing data treatment then the conclusions made from the remaining data are possibly biased and are not likely to represent the original sample accurately. Moreover, these missing data treatments are dependent of a model in a given sample size which can differ as a function of the models specified (i.e. a covariate inclusion or exclusion with missing values will consequent in a different or a new sample when any of the LD or PD is made use of). However it is crucial to understand here that the supposition of the mechanism of missing data is not a supposition which can be checked [6,10]. Accordingly, the simple supposition of MCAR and later use of PD or LD is thought to be unsuitable for the statistical analyses. Here Table 2.2 presents the pattern of Pairwise Deletion [12,15].

Table 2.2 Pairwise Deletion Pattern

Subject	Age	Gender	Income
X1	29	~~NAN~~	$55,000
X2	50	M	$36,000
X3	25	F	~~NAN~~

2.2.1.3 Single Imputation

A different procedure of handling with missing data is to impute the missing values. A mean imputation, which is another approach that requires replacing the variables missing values with a mean of that variable's observed values. However, that kind of procedure is not feasible and is not suggested for it results in diminishing the variance of variables for which mean imputation is applied. In such a case, the covariance between the mean imputed variable and other variables in the dataset will be decreased. The model-based approach is another method of single imputation that entails the use of statistical models depending on the observed data for missing values predictions in given variables. In the case of a continuous variable with missing values, multiple regression method is applicable to impute missing value based on the data observed. Still, making use of regression- based imputation can exacerbate the relationship between dataset variables as imputed values are actually a function of other variables within the dataset. This inflation can be reduced by random error introduction in the missing values estimates. Even though the random error inclusion in a regression-based single imputation can aid in reducing the inflation of the variables' relationships, subsequent analyses do not bring into consideration the uncertainty associated with the imputed missing values [15,16,17].

2.2.2 Modern Missing Data Imputation

Likelihood-based methods, multiple imputations, Bayesian and others are considered in modern missing data treatments [18]. Here we deal further with these first two

modern missing data treatments, namely multiple imputation as well as likelihood-based procedures.

2.2.2.1 Multiple Imputations (MI)

A substitute to single-imputation methods is make use of MI. It was initially developed by Rubın in 1987 [2] as a procedure for making public-use data less established on complicated survey data. Its purport for development was also based on the condition that there may be numerous reasons why data are imputed multiple times. Particularly in case of methods of single imputation, the aim is to put instead of missing values with plausible values whose data is logical to assimilate the variables that can facilitate explaining missingness in the Markov Chain Monto Carlo (MCMC) method that has as its goal the generation of numbers at random from a probability distribution [15]. When it comes to continuous variables, it is usually convenient to assume that the variables have a multivariate normal distribution [12]. Let Y be a data matrix composed of the following variables Y_1, Y_2,…Y_J**.** The density function of the multivariate distribution is assumed to be multivariate normal f (Y) ~ N (μ, σ^2). Given the assumption of multivariate normality and that missingness is related to observed variables (MAR) we can use Gibbs sampler to obtain random draws from the joint distribution of the data [10]. Firstly, a variable in the data matrix is sampled from the marginal distribution of that variable given all other variables in the data set $Y_1^{(t+1)} \sim P(Y_1|Y_2^{(t)}, Y_3^{(t)},\ldots., Y_J^{(t)})$. An assumption for an initial value for each of the variables $Y_2^{(t)}$, $Y_3^{(t)}$,…,$Y_J^{(t)}$ is made. Similarly, the next variable in the data set is sampled from the marginal distribution of that variable, given all the variables from the previous step $Y_2^{(t+1)} \sim P(Y_2|Y_1^{(t+1)}, Y_3^{(t)}$, $Y_2^{(t+1)} \sim P(Y_2|Y_1^{(t+1)}, Y_3^{(t)}$. This procedure is repeated for all variables in the dataset yielding a set of point estimates for each variable in $P(Y_{mis}|Y_{obs})$. As $t \rightarrow \infty$, the distribution of Y becomes the target distribution from where imputed values are drawn. Therefore, MCMC is used for random draws of the missing values from $P(Y_{mis}|Y_{obs})$ to be generated. Preceding the inferential analysis of data is MI a missing data treatment. Multiple imputation is occasionally criticized, and many of the criticisms have been addressed by researchers who study missing data techniques. This section will not recap all of

the aforementioned criticism or every argument made in response to those criticisms; instead, we will discuss two criticisms in particular. The first criticism argues that MI simply makes up data due to the fact that MI relies on simulation, and Schafer (1999) argued that it is not an accurate representation of multiple imputation [19]. This criticism would be true for single imputation that treats the missing values as though their values are known. MI, however, simply makes attempts at evaluating the uncertainty of the statistical model as a result of having missing values without knowing what those missing values are. Alternative approaches like single imputation and LD assume that the missing values do not confer any uncertainty. The second criticism argues that by using MI, the user assumes that there is more information or more data than there actually is. When performing inferential analysis on every complete dataset, Rubin's rules can be applied to combine the parameters estimates, whereby the variance and between imputation variance is calculated. As a result, the proportion of missing information associated uncertainty in the parameters estimate both increases. An increase in uncertainty caused by missing data results in a penalization in the certainty of parameter values. Maximum likelihood (ML) estimation is an alternative missing data treatment that is used in along with inferential analysis [20].

2.2.2.2 Maximum likelihood Imputation (ML)

Maximum likelihood estimation is an estimation technique that aims to maximize the likelihood for a given data to have been observed. This technique is conditional on a set of parameters of interest, $L(Y \mid \theta)$, where Y is a data matrix that consists of both observed and $R|\theta,\omega)$. It has been shown that when likelihood-based or Bayesian approaches are used to estimate the parameters of interest under ignorability, a model for the missing data mechanism need not be incorporated [2]. Since the parameters that describe the missing data mechanisms and the parameters of interest are distinct under the ignorability assumption, we can factorize the likelihood function pertaining to the observed data (observed data likelihood) and the missing data mechanism into distinct components. This factorization makes the observed data likelihood proportional to the likelihood function of the parameters given the observed data.

Therefore, inferences regarding the parameters that govern the dependent variable distribution can be made based on the observed data likelihood under MAR. Schafer, J. L. (1997) [21] uses an arbitrary pattern of missingness to illustrate the observed data likelihood function for the example of multivariate normal data as shown below:

$$\prod_{i=1}^{S} \prod_{i \epsilon I(s)} | \Sigma_s^* |^{-1/2} \exp(-1/2(y_i^* - \mu_s^*)^T \Sigma_s^{*-1} (y_i^* - \mu_s^*)$$

Here, *s* is the distinct missing data pattern among the variables, *i* is the number of cases within a distinct missing data pattern (s), y_i^* is the observed data within a case, μ_s^* is the vector of means for the observed variables in a given s, and Σ_s^* is the variance covariance matrix for observed variables in a given s. Schafer notes that a likelihood function like this requires iterative approaches such as EM algorithm as it is complex and difficult to estimate. An EM algorithm consists of two steps: an expectation step and a maximization step. The expectation step is that the algorithm computes the expected value of the parameters of interest. The maximization step is that the expected values generated as the output of the previous step are used to maximize the parameters of interest. Schafer describes the EM algorithm by mentioning that for any data problem where missing data are present, the distribution of the complete data (Y) can be written as $P(Y|\theta) = P(Y_{obs}|\theta)\ P(Y_{mis}|Y_{obs})$.This can then be written as a likelihood function in terms of θ as $l(\theta|Y\ ) = l(\theta|Y_{obs}) + \log(P(Y_{mis}|Y_{obs},\theta)) + c$, where c is an arbitrary constant. The predictive distribution for Y_{mis}, $P(Y_{mis}|Y_{obs}|\theta)$ cannot be calculated because Y_{mis} is unobserved. Instead, $l(\theta|Y)$ is averaged over log $(P(Y_{mis}|Y_{obs},\theta))$ given a preliminary estimate of $\theta(\theta^{(t)})$. This average is calculated in the EM algorithm's E step and yields $\int(\theta|Y\ )P(Y_{mis}|Y_{obs}, \theta^{(t)})dY_{mis}$. The value of θ that maximizes the function above ($\theta^{(t+1)}$) is computed in the EM algorithm's M step. This new estimate generated in the M step is then for the next E step and this process is repeated until the observed data likelihood converge, thereby yielding maximum likelihood estimates of the parameters of interest.

2.3 Literature Review

In recent years, research has been performed regarding the missing data problem and their handling algorithms, where different algorithms have been used to impute

missing value in a data set. Here we review different imputation approaches used in the literature.

Table 2.3 Summary of the study [22]

The ratio of Missing Values in Dataset			
Method	5% MV	10% MV	20% MV
SVR	1.390-1.352	1.559-1.524	1.897-1.815
Step Reg	1.677-1.558	2.205-1.965	2.481-2.292
ANN	1.829-1.693	2.384-2.120	2.880-2.581
Stepwise Reg & ANN	1.596-1.684	2.059-1.905	2.435-2.255
Mean Subset	4.975	4.975	5.410
Simple Lin Reg	2.582	2.508	2.811

In [22], Richman, Trafalis & Adrianto use different machine learning methods including Support Vector Regression (SVR) and Artificial Neural Network (ANN) for testing against standard imputation methods (e.g., multiple regression), casewise deletion, mean substitution and simple regression. All these methods were applied to impute MVs on the 400 variables of climatological data- a large number of observations. This is followed by presentation of the MSE to assess each technique's efficiency. The results indicated that the non-iterative methods, such as mean substitution and casewise deletion caused a large error whereas the iterative imputation has considerably lower errors. Iterative techniques and SVRs are most promising with regards to reducing error as shown in Table 2.3.

In [23], Paul and Sill used a Pattern Similarity Matching (PSM) algorithm to estimate the value of microarray experimental data. Here, MVs are imputed using a fuzzy similarity measure by recognizing the genes having similar characteristics to those of a gene with missing values. The missing values are predicted as well as optimized. The MV imputing method is used for estimation accuracy of that method as compared to KNN. The result shows that the PSM method outperforms the KNN based method. The NRMSE results are shown in Table 2.4.

Table 2.4 Summary of the study [23]

GEN	The ratio of Missing Values in Dataset			
GDS2771	Method	5% MV	10% MV	20% MV
	PSM	0.08	0.09	0.1
	KNN	0.15	0.25	0.35
GSE4115	PSM	0.02	0.08	0.1
	KNN	0.2	0.25	0.5

In [24], Susianto, Notodiputro, Kurnia & Wijayanto used three imputation algorithms: EM algorithm, Markov Chain Monte Carlo (MCMC) method and the Yates method. These algorithms were used on per-capita expenditure data at a sub-districts level in Central Java to estimate MV and assess the result of estimating algorithms through comparison of the Mean Absolute Error (MAE) and MSE. The result state that MSE and MAE generated using the Yates method are lower than the MSE and MAE yielded from both the MCMC method and the EM algorithm. As a result, the Yates method has been recommended for imputation of the missing values for per capita expenditure at the sub-district level. Table 2.5 shows the results of these 3 algorithms.

Table 2.5 Summary of the study [24]

Districts	Yates		EM		MCMC	
	MSE	MAE	MSE	MAE	MSE	MAE
Banyumas	3.5×10^4	490.03	5.6×10^4	527.71	6.6×10^4	531.80
Pati	2.7×10^4	460.96	3.4×10^4	520.30	3.8×10^4	533.48
Semarang	4.9×10^4	635.33	5.7×10^4	632.76	9.7×10^4	769.88
Brebes	1.8×10^4	456.01	2.1×10^4	460.80	2.2×10^4	450.62
Kota Semarang	15.5×10^4	885.61	18.0×10^4	959.71	18.2×10^4	915.84

2.4 K Nearest Neighbor algorithm (KNN)

A simple and widely used machine learning agorithm is the KNN algorithm, a nonparametric, supervised machine learning algorithm used for regression and classification. It is the most widely used machine learning algorithms owing to its

relatively high performance and intuitive nature. A nonparametric test is a technique for estimating classes without making any assumptions. The nonparametric nature of KNN is useful for checking whether its parametric counterpart model works well. It is also needed to determine the constant k before we implement the learning process, where k is a constant that represents the number of neighbors when considering the class and the best value for k is dependent on the data. classification noise is generally reduced by large k values and also result in less distinct class boundaries. Various heuristic methods can be used to select a good k value. It is useful to set k as an odd number in binary classification problems, as this avoids tied votes [25]. Figure 2.2 depicts the KNN mechanism example.

Here the example examines a situation where a test sample (indicated by a red star) needed to be classified into either Class A (yellow circles) or Class B (purple circles). Majority voting was used for classification. As an example, if k = 3 (indicated by the inner dashed line circle) the test sample is assigned to the class B as there is only 1 yellow circle and 2 purple circles within the inner circle. When k = 6 (indicated outer dashed line circle) it is assigned to class A (4 yellow circles vs. 2 purple circle inside the outer circle).

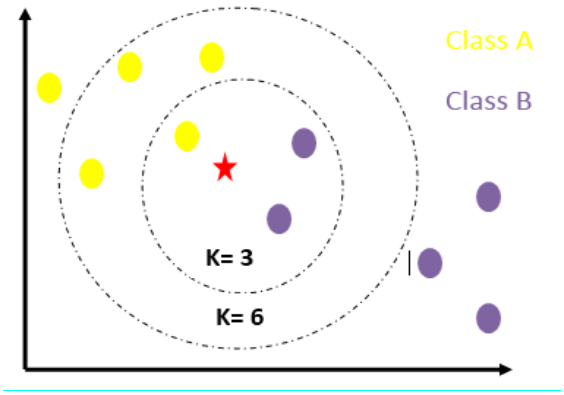


Figure 2.2 KNN Mechanism

If k = 1, then the test sample is just assigned to the class of its nearest neighbor. K nearest neighbors calculated by a distance function. Euclidean distance is used here which is a typically used distance function. The n-dimensional Euclidean distance between a and b can be written as:

$$E(a.b) = \sqrt{\sum_{i\epsilon D} (x\,ai - x\,bi)^2}$$

- E (a, b) is the distance between the cases a and b.
- X_{ai} and X_{bi} are the values of attribution in cases a and b.
- D is the set of attributes with non-missing values in both mentioned cases.

2.5 Singular Value Decomposition (SVD)

In this method, we can approximate all attributes of the data set by using singular value decomposition to form mutually orthogonal expression patterns. These are later combined linearly for the attribute approximation. In our case, the principal components of the data values' matrix, also called eigenvalues, are identical to the patterns mentioned above.

$$A_{m\times m}=U_{m\times m} \sum_{m\times m} V^{T}_{n\times n}$$

Matrix V^T now consist of eigenvalues, whose presentation to the expression in the eigenspace is by equivalent eigenvalues on the diagonal of the matrix $\sum$. We then spot the most important eigenvalues by classifying the eigenvalues depending on their propositional eigenvalue. It has been argued that numerous important eigenvalues are enough to explain expression data; however, empirical determination is required to determine the exact fraction of eigenvalues which is the best for optimal estimation.

A missing value marked by j in a row, i, regressed against k eigenvectors is estimated after selecting the k number of most significant eigenvectors from V^T. The coefficients of regression are used to reconstruct j out of a linear k eigenvalue combination. The j th value of example i and the j th value of the K eigenvalues are not used in determining these regression coefficients. SVD can only be used when dealing with complete matrices. As a result, the row average of all of the missing values in A' is substituted, followed by acquiring the final estimate via utilization of a Regularize Expectation -Maximization (REM) method. Hence, each missing value in A' is estimated as described above, and the process is repeated on the newly

generated matrix up until the total change within the matrix is reduced to below the empirically determined one (this is referred to as stagnation tolerance in the EM algorithm by the authors). All other parameters in the EM algorithm are identical for both algorithms [26].

CHAPTER 3

METHODOLOGY

3.1 Introduction

Missing values are extremely undesirable in data mining, machine learning, and other information systems. One may use other methods in order to deal with missing values in datasets such as ignoring, deleting, zero or mean instead of imputation methods. However, there are some significant disadvantages of using these methods like the loss of efficiency due to discarding incomplete observations and biases in estimates in case data are missing a systematic manner. These disadvantages reduce data quality. High-quality data mining results can be obtained only with high-quality data. Therefore, missing values should be estimated properly to increase data quality. Missing values typically occur due to various reasons like sensor faults, a lack of response in scientific experiments, faulty measurements, data transfer problems in digital systems or because of respondents who are unwilling to respond to survey questions. In scientific research, unfortunately sometimes data for some variables in the database which has to be properly analyzed may be missing. In such a case if the missing values are not treated correctly, they may decrease or even jeopardize the validity of the entire research.

This chapter presents and illustrates the methodology of the proposed technique used in the current thesis to eradicate the mentioned problem. In the proposed methodology, two algorithms, namely KNN and SVD, are proposed in hybrid form to impute missing values in data sets. The main reason to use this form in imputation is to decrease the error rate that is caused by missing values and benefit from properties of these algorithms in dealing with high dimensional data which has missing values such as the Yeast data set that we use in our experiment. This chapter firstly focuses on the presentation of the thesis framework. After that, we discuss in

detail the different stages of the framework. In the end, performance metrics are discussed thoroughly.

3.2 Research Framework

Figure 3.1 explains that the research framework contains several stages, beginning with the problem research and then ending with the performance metric that is used for evaluating the proposed technique.

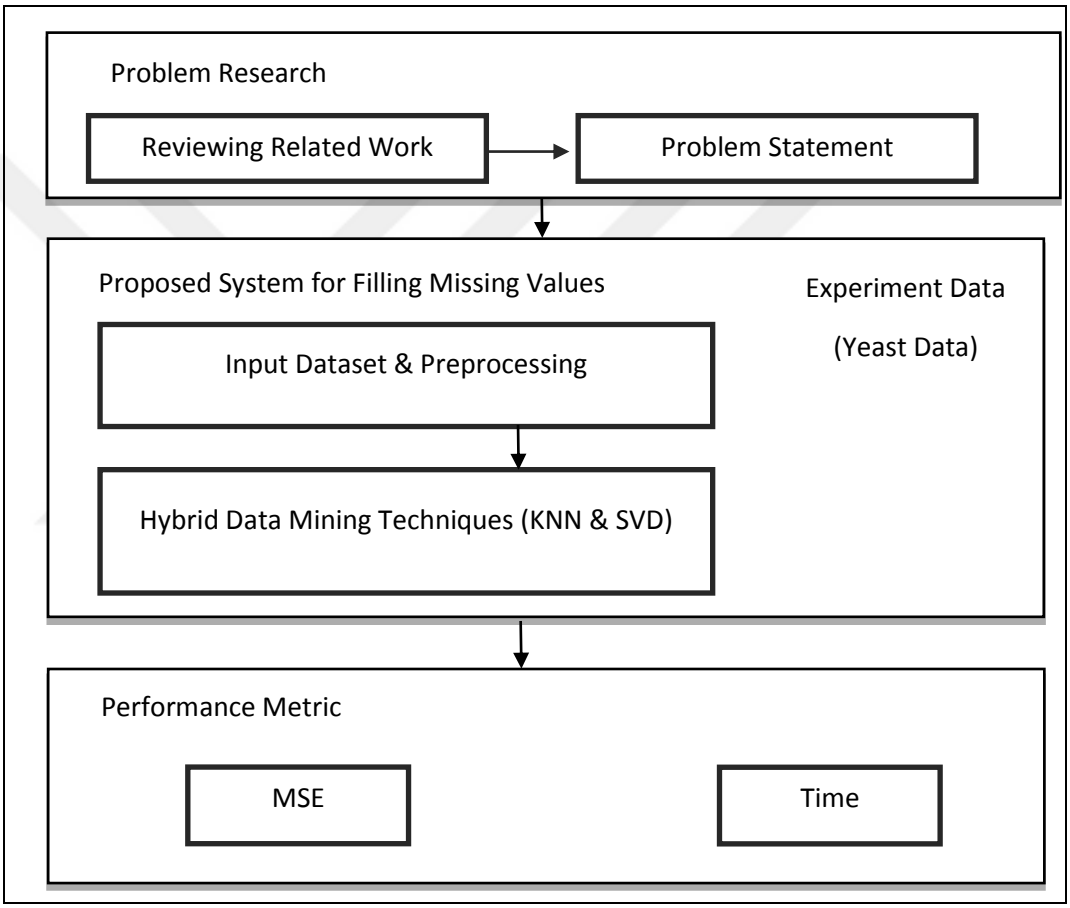


Figure 3.1 Research Framework

3.3 Proposed System for Filling Missing Values

The structure of the proposed system consists of certain stages. Each stage has specific functions and all the functions are explained in detail in the following subsections. Figure 3.2 describes the block diagram of the proposed system for filling missing values.

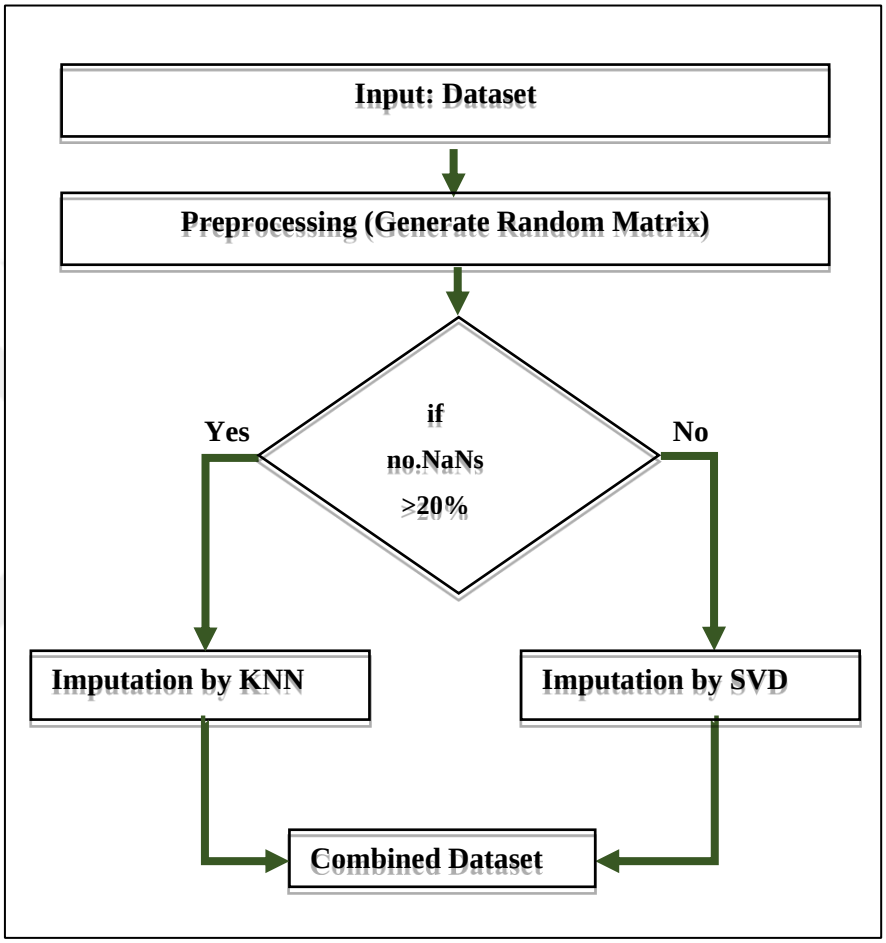


Figure 3.2 Proposed System

- **First Step (Input Dataset):** The first step of the block diagram is related to input data that is used in the experiment. The Yeast dataset which does not contain any missing value is used in this thesis. The Yeast dataset was taken from UCI Machine Learning Repository, which is an institute of Molecular and Cellular Biology in Osaka University related to Predicting the Cellular Localization Sites of Proteins. The Yeast dataset contains 1484 rows with eight attributes as summarized in Table 3.1. Sample data of the Yeast dataset is shown in Table 3.2.

Table 3.1 Structure of Yeast Data

Attributes	Missing Values	Task	Characteristics	Instance
8	No	Classification	Real	1484

Table 3.2 Sample of Yeast Data

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
0.5	0.3	0.4	0.2	0.7	0.1	0.01	0.14
0.12	0.11	0.55	0.45	0.66	0.76	0.33	0.32
0.2	0.29	0.41	0.34	0.55	0.66	0.77	0.87
0.86	0.83	0.52	0.23	0.43	0.87	0.99	0.87
0.5	0.3	0.4	0.2	0.7	0.1	0.01	0.14
0.12	0.11	0.55	0.45	0.66	0.76	0.33	0.32
0.2	0.29	0.41	0.34	0.55	0.66	0.77	0.87

- **Second Step (Preprocessing Dataset):** This step prepares the dataset for the imputation techniques using 1000 random location generator that is mean about 10% of MVs in all dataset. This generator selects random values from the dataset and replaces original values with NaN. Firstly, it generates a random matrix using a random permutation function which contains two types of values 0 and 1, where 1 represents the missing values and 0 represents non-missing values as shown in Table 3.3.

  Then, it searches for the values of 1's and 0's in Table 3.3 and compares it to the corresponding values in Table 3.2. Consequently, it generates a new Table like Table 3.4 that contains NaN instead of 1's as shown in Table 3.3.

Similarly, in Table 3.2 actual values are put instead of 0's as shown in Table 3.3.

Table 3.3 Random Missing Values Generation

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	0	0	0
1	1	0	0	0	1	0	0
0	1	0	0	1	0	0	1
0	0	0	1	0	0	0	0

Table 3.4 Dataset with NaNs

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
0.5	0.3	0.4	0.2	NaN	0.1	0.01	0.14
NaN	0.11	0.55	0.45	0.66	NaN	0.33	NaN
0.2	NaN	0.41	0.34	0.55	0.66	NaN	0.87
0.86	0.83	NaN	0.23	0.43	0.87	0.99	0.87
NaN	NaN	0.4	0.2	0.7	NaN	0.01	0.14
0.12	NaN	0.55	0.45	NaN	0.76	0.33	NaN
0.2	0.29	0.41	NaN	0.55	0.66	0.77	0.87

- **Third Step (Calculate the Percentage of Missing Values for each Row):** In this step, the KNN or SVD technique is used based on a threshold value. Therefore, the dataset is split into two parts according to the percentage of threshold in each row to decide which algorithm is to be used.

  A. The first group refers to data that is extracted from Table 3.4 where each row in the table has more than 20% of missing value.

  B. The second group refers to a same table of data that extracted from Table 3.4 where the percentage of missing values in each row is less than 20%.

Table 3.5 Splitting Process

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Threshold	Method
0.5	0.3	0.4	0.2	NaN	0.1	0.01	0.14	< 20 %	SVD
NaN	0.11	0.55	0.45	0.66	NaN	0.33	NaN	> 20 %	KNN
0.2	NaN	0.41	0.34	0.55	0.66	NaN	0.87	> 20 %	KNN
0.86	0.83	NaN	0.23	0.43	0.87	0.99	0.87	< 20 %	SVD
NaN	NaN	0.4	0.2	0.7	NaN	0.01	0.14	> 20 %	KNN
0.12	NaN	0.55	0.45	NaN	0.76	0.33	NaN	> 20 %	KNN
0.2	0.29	0.41	NaN	0.55	0.66	0.77	0.87	< 20 %	SVD

Fourth Step (Imputation) : in this step the first group that its rows have more than 20% of MVs are imputed using KNN algorithm and the second group that its rows have less than 20% of MVs are imputed using SVD.

Fifth Step (Evaluation) :in this step, the mean squared error **(MSE)** and **Time** are used to evaluate the result of the proposed method.

3.4 Imputation Techniques

Two data mining techniques are used in this stage for data imputation. The first one is the KNN algorithm, which is used for estimating and substituting the missing values when the percentage of missing values is (greater than 20% for a row) in our study. Whereas the second one is the SVD algorithm, which is used for estimating and substituting the missing values when the rate of missing values is small (less than 20% for a row). In the current thesis, we use KNN and SVD in a hybrid form to impute missing values after testing KNN and SVD alone and after testing the hybrid case with different percentage of threshold for missing values (20%, 40%, 50% and 70%) to determine the best threshold that will give us the minimum error.

3.4.1 K-Nearest Neighbor (KNN) Algorithm

The first algorithm used for data imputation is the K-nearest neighbor (KNN) algorithm which is based on three criteria:

- An integer K decides how many neighbors will be taken for assessing the missing values in each iteration.
- Training data is represented by rows with complete attribute values.
- Closeness shows the distance of a specific row from the others. The key method for calculating it is the Euclidean Distance.

KNN imputation method replaces NaN values with the corresponding values from the nearest neighbor column using Euclidean Distance if the number of missing values is one. However, when the nearest neighbor column is also NaN, then the next nearest neighbor column is used. Each unknown sample NaN that has a missing value (s), computes the distance between the missing input recorded with all the training data depending on the Euclidean Distance metrics. Then the K's smallest distance that is in the corresponding recognized samples has to be found. After that, the missing value has to be replaced by the equivalent attribute value of the most similar complete record. The procedure has to be repeated until all missing values are imputed. The KNN imputation technique is described in the following Algorithm steps [27] [28].

KNN Algorithm Steps
Input: Testing data (with missing values), Training data (without missing values), K represents the number of neighbors.
Output: Testing data (with imputed values)
Begin For each missing input recorded in testing data: $Step_1$: Compute the distance between the row values of missing input recorded and the row values of all the training data rows using Euclidean Distance function. $Step_2$: Find the rows in the training data giving K the smallest distance and replacing it by the equivalent attribute value of the most similar complete record. $Step_3$: Exchange the missing value with the value of the nearest neighbor or select a value by majority voting when the nearest neighbor is NaN. End

3.4.1.1 Case Study for KNN Imputation

This section describes complete two examples of KNN imputation.

Example 1: the data given in Table 3.6 consists of acid durability, strength and quality attributes and has a missing value.

Table 3.6 Testing Data (with missing values)

Acid durability	Strength	Quality
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good
3	7	**NaN**

Table 3.7 Training Data (without missing values)

Acid durability	Strength	Quality
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Then the procedure to be followed is explained step by step below:

- Suppose k=3
- Calculate the Euclidian distance between the row which has NaN value with all others:

$$d(5,1) = \sqrt{(7-3)^2 + (7-7)^2} = 4$$

$$d(5,2) = \sqrt{(7-3)^2 + (4-7)^2} = 5$$

$$d(5,3) = \sqrt{(3-3)^2 + (4-7)^2} = 3$$

$$d(5,4) = \sqrt{(1-3)^2 + (4-7)^2} = 3.6$$

- Rank distances from smallest to greatest:

d (5,1) = $\sqrt{7-3\ ^2+\ 7-7\ ^2}$ =4→ Rank3

d (5,2) = $\sqrt{7-3\ ^2+\ 4-7\ ^2}$=5 → Rank4

d (5,3) = $\sqrt{3-3\ ^2+\ 4-7\ ^2}$=3 → Rank1

d (5,4) = $\sqrt{1-3\ ^2+\ 4-7\ ^2}$=3.6 → Rank2

- Take the first 3 since K is defined as:

d (5,3) = $\sqrt{3-3\ ^2+\ 4-7\ ^2}$ =3→ Rank1→ Good

d (5,4) = $\sqrt{1-3\ ^2+\ 4-7\ ^2}$ =3.6 → Rank2→ Good

d (5,1) = $\sqrt{7-3\ ^2+\ 7-7\ ^2}$ =4 → Rank3→ Bad

- Use majority of the nearest neighbors (2 Good, 1 Bad) for imputing the missing value:

3	7	Good

Table 3.8 Imputation Data for Example 1

Acid durability	Strength	Quality
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good
3	7	**Good**

Example 2: Table 3.9 contains data from Table 3.5. This will be processed using KNN algorithm.

Table 3.9 Testing Data for Example 2 (with missing value)

X	Y	Z	W
NaN	0.11	0.55	0.45
0.2	NaN	0.41	0.34
NaN	NaN	0.4	0.2
0.12	NaN	0.55	0.45

The procedure to be followed is given step by step below:

- Suppose k=3
- Calculate the Euclidian distance between the row which has NaN value with all others:

- The distance between the first row and the others

$d(1,2) = \sqrt{(0.41-0.55)^2 + (0.34-0.45)^2} = 0.12$

$d(1,3) = \sqrt{(0.4-0.55)^2 + (0.2-0.45)^2} = 0.44$

$d(1,4) = \sqrt{(0.55-0.55)^2 + (0.45-0.45)^2} = 0$

- The distance between the second row and the others

$d(2,1) = \sqrt{(0.55-0.41)^2 + (0.45-0.34)^2} = 0.46$

$d(2,3) = \sqrt{(0.41-0.4)^2 + (0.34-0.2)^2} = 0.14$

$d(2,4) = \sqrt{(0.12-0.2)^2 + (0.55-0.41)^2 + (0.54-0.34)^2} = 0.22$

- The distance between the third row and the others

$d(3,1) = \sqrt{(0.55-0.4)^2 + (0.45-0.2)^2} = 0.29$

$d(3,2) = \sqrt{(0.41-0.4)^2 + (0.34-0.2)^2} = 0.14$

$d(3,4) = \sqrt{(0.55-0.4)^2 + (0.45-0.2)^2} = 0.29$

The distance between the fourth row and others

$$d(4,1) = \sqrt{(0.55-0.55)^2 + (0.45-0.45)^2} = 0$$

$$d(4,2) = \sqrt{(0.12-0.2)^2 + (0.41-0.55)^2 + (0.34-0.45)^2} = 0.19$$

$$d(4,3) = \sqrt{(0.4-0.55)^2 + (0.2-0.45)^2} = 0.29$$

- Rank the distances from smallest to greatest:
  - For the first row

$$d(1,4) = \sqrt{(0.55-0.55)^2 + (0.45-0.45)^2} = 0 \rightarrow \text{Rank1 (shortest distance)}$$

$$d(1,2) = \sqrt{(0.41-0.55)^2 + (0.34-0.45)^2} = 0.12 \rightarrow \text{Rank2}$$

$$d\ 1,3 = \sqrt{(0.4-0.55)^2 + (0.2-0.45)^2} = 0.44 \rightarrow \text{Rank3}$$

  - For the second row

$$d(2,3) = \sqrt{(0.41-0.4)^2 + (0.34-0.2)^2} = 0.14 \rightarrow \text{Rank1 (shortest distance)}$$

$$d(2,4) = \sqrt{(0.12-0.2)^2 + (0.55-0.41)^2 + (0.54-0.34)^2} = 0.22 \rightarrow \text{Rank2}$$

$$d(2,1) = \sqrt{(0.55-0.41)^2 + (0.45-0.34)^2} = 0.46 \rightarrow \text{Rank3}$$

In this row we will choose the majority voting because we have more than one missing value in the same column, so we will discard the rank 1 since there is NaN value in the nearest neighbor (column 3) as well as discard rank 2 for the same reason. However, we will choose the rank 3 and put the equivalent value in the first row instead of the missing value.

  - For the third row

$$d(2,3) = \sqrt{(0.41-0.4)^2 + (0.34-0.2)^2} = 0.14 \rightarrow \text{Rank1 (shortest distance)}$$

$$d(3,2) = \sqrt{(0.55-0.4)^2 + (0.45-0.2)^2} = 0.29 \rightarrow \text{Rank2}$$

$$d(3,4) = \sqrt{(0.55-0.4)^2 + (0.45-0.2)^2} = 0.29 \rightarrow \text{Rank2}$$

Here, having the same case as the second row, we will discard the first row and use the majority voting to choose the equivalent value for the nearest neighbor and then choose the rank 2 which is related to the first row and then use their value.

- For the fourth row

d (4,1) = $\sqrt{0.55-0.55\ ^2+\ 0.45-0.45\ ^2}$ = 0→ Rank1 (**shortest distance)**

d (4,2) = $\sqrt{0.12-0.2\ ^2+\ 0.41-0.55\ ^2+\ 0.34-0.45\ ^2}$ = 0.19 Rank2

d (4,3) = $\sqrt{0.4-0.55\ ^2+\ 0.2-0.45\ ^2}$ = 0.29→ Rank3

Finally, imputing missing value by replacing the NaN value with the equivalent value of the cell which has the shortest distance.

Table 3.10 Imputed Data for Example 2

X	Y	Z	W
0.12	0.11	0.55	0.45
0.2	0.11	0.41	0.34
0.2	0.11	0.4	0.2
0.12	0.11	0.55	0.45

3.4.2 Singular Value Decomposition (SVD) Algorithm

The second technique used is the Singular Value Decomposition SVD. For starting imputation processes we need to select the number of iterations. The Singular Value Decomposition of a matrix A is UΣV (the number of singular value decomposition must not be larger than matrix dimension), where U and V are unitary matrices and Σ is a diagonal matrix with positive real elements (SV of the covariance matrix, while A of the eigenvalues is of the matrix AA^t or A^tA).

For missing value imputation, the first phase is scanning the data set, then categorizing the whole value to testing data (observations which have and don't have missing values) and training data (observations which have missing value). After

that, we need to get data out from the missing dataset circularly and record the location of lost attributes in the acquired missing value. Mean imputation should be applied as a processing step (because SVD deals with complete data to improve the imputed value by mean). For better imputation, the process can be iterated to a certain number of times [29]. The SVD process is described in Algorithm Steps.

SVD Algorithm Steps
Input: Testing data (with missing values), Training data (without missing values), nSV = number of singular values to use nIt = number of iterations to apply for improving the result, V^t is the transpose of V.
Output: Testing data (with imputed values)
Begin Step1: Determine the position of the missing values Step2: Imputation by Mean Step3: i=1 Step4: Compute the three matrixes U, ∑ and Vt. Step5: Calculate the multiplication of the three matrixes, where new SV= U* ∑*Vt Step6: for all new SV copy only, the desired value which is the equivalent position with missing value and put them in the imputed matrix by the mean. Step7: i=i+1 Step8: if (i < nIt) go to step 4 End

3.4.2.1 Case Study for SVD Imputation

Consider a **Y** matrix's size is 2*3 with two random missing values

$$\mathbf{Y}= \begin{matrix} 4 & 0 & \text{NaN} \\ \text{NaN} & 0 & 4 \end{matrix}$$

Initially, these missing values (a_{13}, a_{21}) are imputed using a mean algorithm**,** thereby providing a completed matrix A.

$$\mathbf{A}= \begin{matrix} 4 & 0 & 2 \\ 2 & 0 & 4 \end{matrix}$$

The second step is for applying SVD of A, $U\sum V^T$ to justify the values that are imputed in the first step; this step will be repeated for more than one iteration to choose and which will give us the minimum error. The following steps of the SVD algorithm are meant for the first iteration.

Step$_1$: Compute the transpose of A and A^TA

Since then

$$A^T = \begin{matrix} 4 & 2 \\ 0 & 0 \\ 2 & 4 \end{matrix} \qquad A^TA = \begin{matrix} 4 & 2 \\ 0 & 0 \\ 2 & 4 \end{matrix} \begin{matrix} 4 & 0 & 2 \\ 2 & 0 & 4 \end{matrix}$$

$$A^TA = \begin{matrix} 20 & 16 \\ 16 & 20 \end{matrix}$$

Step$_2$: Determine the eigenvalues of A^TA and sort these in descending order, in absolute sense. Square roots that obtain the singular values of A.

$$A^TA-ג I = \begin{matrix} 20-ג & 16 \\ 16 & 20-ג \end{matrix} = (20-ג)(20-ג)-(256) = 0$$

Characteristic equation $\rightarrow ג^2 + 40ג + 144 = 0$

The quadratic equation gives two values

In decreasing order, these are $\rightarrow |-36| > |-4|$

Eigenvalues $\rightarrow ג_1 = 4,\ ג_2 = 36$

Singular values $\rightarrow \sigma_1 = \sqrt{4} = 2,\ \sigma_2 = \sqrt{36} = 6$

Step $_3$: - Construct diagonal matrix S by placing singular value in descending order along its diagonal. Compute its inverse, S^{-1}.

$$S = \begin{matrix} 6 & 0 \\ 0 & 2 \end{matrix} \qquad S^{-1} = \begin{matrix} 0.1 & 0 \\ 0 & 0.5 \end{matrix}$$

Step 4: - Use the ordered eigenvalues from step 2 and compute the eigenvectors of A^TA. Place the eigenvectors along the columns of V and compute its transpose V^T.

For $\lambda_1 = 4$

$AA^T.\lambda I = \begin{matrix} 20-4 & 16 \\ 16 & 20-4 \end{matrix} = \begin{matrix} 16 & 16 \\ 16 & 16 \end{matrix}$

$(A^TA-\lambda I)\ X_1 = 0$

$$\begin{matrix} 16 & 16 \\ 16 & 16 \end{matrix} \begin{matrix} X1 \\ X2 \end{matrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

$16\ X_1 + 16X_2 = 0$

$16\ X_1 + 16X_2 = 0$

Solving for X_2 for either equation: $X_2 = -X_1$

Dividing by length,

$L = \overline{X1} + \overline{X2} = X1\ \overline{2}$

$V_1 = \begin{matrix} \frac{X1}{L} \\ -\frac{X1}{L} \end{matrix} = \begin{matrix} \frac{1}{\overline{2}} \\ -\frac{1}{\overline{2}} \end{matrix} = \begin{matrix} 0.7 \\ -0.7 \end{matrix}$

For $\lambda_2 = 6$

$AA^T.\lambda I = \begin{matrix} 20-36 & 16 \\ 16 & 20-36 \end{matrix} = \begin{matrix} -16 & 16 \\ 16 & -16 \end{matrix}$

$(A^TA-\lambda I)\ X_1 = 0$

$$\begin{matrix} -16 & 16 \\ 16 & -16 \end{matrix} \begin{matrix} X1 \\ X2 \end{matrix} = \begin{matrix} 0 \\ 0 \end{matrix}$$

$-16\ X_1 + 16X_2 = 0$

$16\ X_1 - 16X_2 = 0$

Solving for X_2 for either equation: $X_2 = X_1$

Dividing by length,

$L = \overline{X1} + \overline{X2} = X1\ \overline{2}$

$V_2 = \begin{matrix} \frac{X1}{L} \\ \frac{X1}{L} \end{matrix} = \begin{matrix} \frac{1}{\overline{2}} \\ -\frac{1}{\overline{2}} \end{matrix} = \begin{matrix} 0.7 \\ 0.7 \end{matrix}$

$V = X1\ \ X2 = \begin{matrix} 0.7 & 0.7 \\ -0.7 & 0.7 \end{matrix}$ $\qquad V^T = \begin{matrix} 0.7 & -0.7 \\ 0.7 & 0.7 \end{matrix}$

Step$_5$: - Compute U as $U = AVS^{-1}$. To complete the proof, compute the full SVD using $A = U\sum V^T$.

$$U = \begin{matrix} 4 & 0 & 2 & 0.7 & 0.7 & 0.1 & 0 \\ 2 & 0 & 4 & -0.7 & 0.7 & 0 & 0.5 \end{matrix}$$

$$U = \begin{matrix} 4 & 0 & 2 & 0.07 & 0.3 \\ 2 & 0 & 4 & -0.07 & 0.35 \end{matrix}$$

$$U = \begin{matrix} 0.98 & 0 & 1.54 \\ 0.42 & 0 & 1.26 \end{matrix}$$

$A = U\sum V^T = \begin{matrix} 0.98 & 0 & 1.54 & 6 & 0 & 0.7 & -0.7 \\ 0.42 & 0 & 1.26 & 0 & 2 & 0.7 & 0.7 \end{matrix}$, Where $\sum$ equal the value of singular value

$$A = \begin{matrix} 6 & 0 & 12 \\ 1 & 0 & -0.3 \end{matrix}$$

The third step is replacing the new values (12, 1) instead of the values that are imputed in the second step (2, 2) to obtain a new matrix A' and compute MSE for A'

$$A' = \begin{matrix} 4 & 0 & 12 \\ 1 & 0 & 4 \end{matrix}$$

Mean Squared Error (MSE) is an important criterion that is used for measuring the performance of an estimator and is quite crucial for relaying on the concepts of precision, bias, and accuracy during the statistical estimation. The formula for mean squared error is given below:

$$\boldsymbol{MSE = \frac{1}{n} \sum_{i=1}^{n} X_i' - X^{\,2}}$$

where *X'i* is the vector denoting values of n number of predictions. *Xi* is a vector representing n number of original values.

MSE A' = 5.5

The process is then iterated (return to the second step) using A' until stability is achieved out in the imputations. For instance, iterations are continued till the differences between the estimated values in the current iteration and those in the

previous iteration are smaller than the prespecified small value; in our example 0.1 is the threshold of MSE. In this step SVD calculation is applied on the A' to compute three matrixes U, $\sum$ and V^T.

$$U=\begin{matrix}0.9 & -0.3\\ 0.3 & 0.9\end{matrix},\ \sum=\begin{matrix}13.3 & 0 & 0\\ 0 & 0.3 & 0\end{matrix},\ V^T=\begin{matrix}0.3 & 0 & 0.9\\ -0.9 & 0 & 0.3\\ 0 & 1 & 0\end{matrix}$$

NEW SV = $\begin{matrix}0.11 & -0.09 & 0\\ 0.39 & 0.27 & 0\end{matrix}$ then return to the third step to compute A" and MSE for this second iteration.

A"= $\begin{matrix}4 & 0 & 0\\ 0.39 & 0 & 4\end{matrix}$

MSE= 6.3

Return to second step for SVD calculation using A"

$$U=\begin{matrix}0.9 & -0.09\\ 0.09 & 0.9\end{matrix},\ \sum=\begin{matrix}4.01 & 0 & 0\\ 0 & 0.3 & 0\end{matrix},\ V^T=\begin{matrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{matrix}$$

NEW SV = $\begin{matrix}3.6 & -0.27 & 0\\ 3.6 & 0.27 & 0\end{matrix}$ then return to the third step to compute A" and MSE for this third iteration.

A'"= $\begin{matrix}4 & 0 & 0\\ 3.6 & 0 & 4\end{matrix}$

MSE = 1.6

Return to second step for SVD calculation using A'"

NEW SV = $\begin{matrix}2.4 & 1.5 & -2\\ 3.9 & 2.4 & 1.2\end{matrix}$ then return to the third step to compute A"" and MSE for this fourth iteration.

A""= $\begin{matrix}4 & 0 & -2\\ 3.9 & 0 & 4\end{matrix}$

MSE = 0.8

Return to second step for SVD calculation using A""

NEW SV= $\begin{matrix} 2.9 & 0 & -1.5 \\ 2.7 & 0 & 2.9 \end{matrix}$ then return to the third step to compute A''''

and MSE for this fifth iteration.

A''''= $\begin{matrix} 4 & 0 & -1.5 \\ 2.7 & 0 & 4 \end{matrix}$

Return to second step for SVD calculation using A''''

NEW SV = $\begin{matrix} 3.6 & 0 & -1.2 \\ 2.2 & 0 & 3.6 \end{matrix}$ then return to the third step to compute A''''

and MSE for this sixth iteration.

A''''= $\begin{matrix} 4 & 0 & -1.2 \\ 2.2 & 0 & 4 \end{matrix}$

MSE = 0.1

After six iterations, the value of error reaches to stability case because the value of error stays at 0.1, then we choose the iteration which has the minimum error in our example. Therefore, A'''' will be the new imputed matrix instead of Y matrix [29][30].

CHAPTER 4

EXPERIMENT ENVIRONMENT AND RESULTS

4.1 Introduction

This chapter displays the results of the suggested hybrid method using KNN and SVD algorithms. In the end, it evaluates the result of the imputation method using MSE and time.

4.2 Implementation Environment

For testing data, the Yeast dataset consisting of 1484 rows and eight attributes is selected for this study. The Yeast dataset is available online https://archive.ics.uci.edu/ml/datasets/Yeast . Fig 4.1 shows a sample from the Yeast data. The proposed system is implemented using MATLAB 2014. In addition, the proposed system works under the Windows 7 Operating System and uses a platform of Intel Core i3 with 4GB RAM.

1484x8 double

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	0.2200
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	0.5300	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	0.1700	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	0	0.5800	0.3400
9	0.5500	0.5000	0.6600	0.3600	0.5000	0	0.4900	0.2200
10	0.4000	0.3900	0.6000	0.1500	0.5000	0	0.5800	0.3000
11	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
12	0.4200	0.3700	0.5900	0.2000	0.5000	0	0.5200	0.2900
13	0.4000	0.4200	0.5700	0.3500	0.5000	0	0.5300	0.2500
14	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
15	0.6600	0.5500	0.4500	0.1900	0.5000	0	0.4600	0.2200

Figure 4.1 Sample of Yeast Data

4.3 Implementation of Proposed System

Two windows of the implemented System are given in Figures 4. 2 and 4. 3.

A- The left-hand side of Figure 4.2 shows the data with threshold more than 20% of missing values and the right-hand side shows data with threshold less than 20% of missing values.

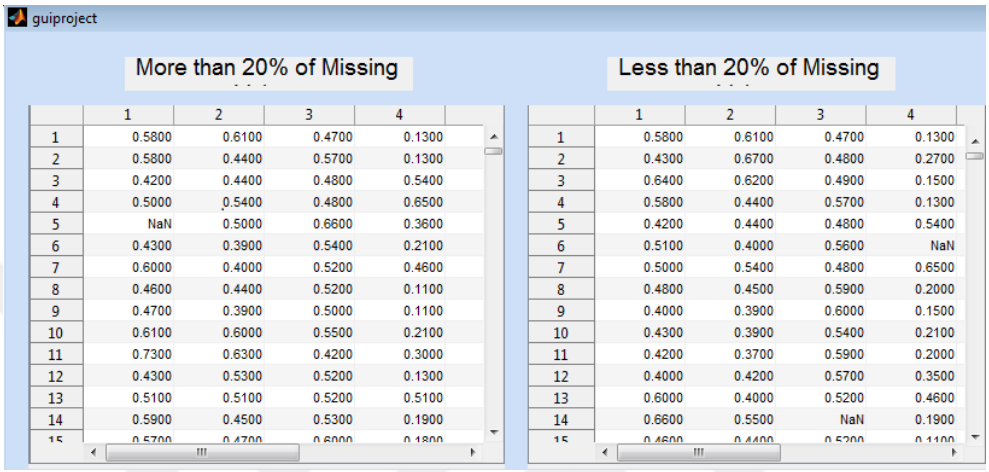


Figure 4.2 Generation of Missing Values

B- The left-hand side of Figure 4.3 is dedicated for data imputed using the KNN method and the right-hand side is dedicated for data imputed using the SVD method.

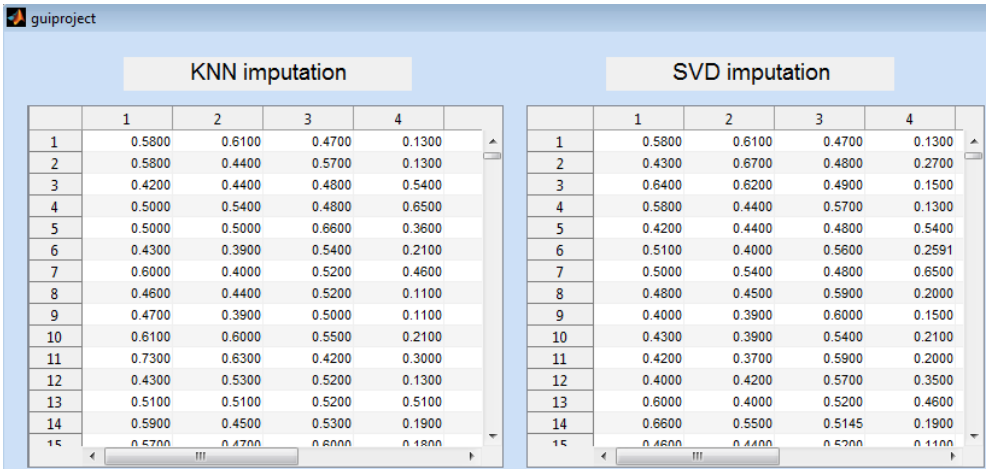


Figure 4. 3 Imputation of Missing Values

4.4 Case Study for Suggested Hybrid Method

The idea of the suggested method is to combine two techniques which are KNN and SVD together. The critical point in the method is to select the correct algorithm during imputation. This can be achieved by using a condition within the suggested algorithm. In this way it allows reaching a minimum error as compared to using KNN or SVD alone. The hybrid method allows both KNN and SVD to work together by means of letting them run simultaneously in the same data. Firstly, the data needs checking and there is a pre-processing needed before the hybrid algorithm is run. Then, a condition is set to check the percentage of a missing value within each row. This suggested hybrid method is explained in 4 steps below:

- **First Step (Input Dataset):** load the Yeast dataset that consisting of 1484 rows and 8 attributes.

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	NaN
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	NaN	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	NaN	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	NaN	0.5800	0.3400
9	NaN	0.5000	0.6600	0.3600	0.5000	0	0.4900	NaN
10	0.4000	0.3900	0.6000	0.1500	0.5000	NaN	0.5800	0.3000
11	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
12	0.4200	0.3700	0.5900	0.2000	0.5000	0	NaN	0.2900
13	0.4000	0.4200	0.5700	0.3500	0.5000	0	NaN	0.2500
14	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
15	0.6600	0.5500	NaN	0.1900	0.5000	0	0.4600	0.2200

Figure 4.4 Random Missing Values Generation

- **Second Step (Preprocessing Dataset):** Generate about 1000 random missing values on the whole Yeast data and calculate the percentage of missing values in each row. In Figure 4.4, the values in some cells are deleted randomly so some rows have more than 20% of missing values and

others have less than 20% of missing values whereas some rows are complete.

- **Third Step (Calculate the Percentage of Missing Values):** in this step the percentage of missing values is calculated for each row in the Yeast dataset where the data separates into two groups. The first group is shown in Figure 4.5 and the second group is shown in Figure 4.6, respectively.

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
3	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
4	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
5	NaN	0.5000	0.6600	0.3600	0.5000	0	0.4900	NaN
6	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
7	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
8	0.4600	0.4400	0.5200	0.1100	0.5000	0	0.5000	0.2200
9	0.4700	0.3900	0.5000	0.1100	0.5000	0	0.4900	0.4000
10	0.6100	0.6000	0.5500	0.2100	0.5000	0	0.5000	0.2500
11	0.7300	0.6300	0.4200	0.3000	0.5000	0	0.4900	0.2200
12	0.4300	0.5300	0.5200	0.1300	NaN	NaN	0.5500	0.2200
13	0.5100	0.5100	0.5200	0.5100	0.5000	0	0.5400	0.2200
14	0.5900	0.4500	0.5300	0.1900	0.5000	0	0.5900	0.2700
15	0.5700	0.4700	0.6000	0.1800	0.5000	0	0.5100	0.2200

Figure 4.5 First Group with Threshold More than 20%

- Figure 4.5 displays the sample of the Yeast dataset after the missing values generation where NaN represent values that are missed. In addition, this figure illustrates the threshold of missing values for each row where some rows do not have missed values and others have more than 20% of missing value (2 or more cells in each row) where these cells will be imputed using the KNN algorithm.
- The second group is shown in Figure 4.6 where the threshold of missing values in some rows are 0 and the others have less than 20% of missing value. These cells will be imputed using the SVD algorithm.

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	NaN
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	NaN	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	NaN	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	NaN	0.5800	0.3400
9	0.4000	0.3900	0.6000	0.1500	0.5000	NaN	0.5800	0.3000
10	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
11	0.4200	0.3700	0.5900	0.2000	0.5000	0	NaN	0.2900
12	0.4000	0.4200	0.5700	0.3500	0.5000	0	NaN	0.2500
13	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
14	0.6600	0.5500	NaN	0.1900	0.5000	0	0.4600	0.2200
15	0.4600	0.4400	0.5200	0.1100	0.5000	0	0.5000	0.2200

Figure 4.6 Second Group with Threshold Less than 20%

- Replaces the NaN cells in Figure 4.7 and Figure 4.8 with real values which have both more and less than 20% of missing values respectively.

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0.5500	0	0	0	0	0	0	0.2200
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0.5000	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0

Figure 4.7 Real Value of NaNs in Figure 4.5

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0.2200
3	0	0	0	0	0	0	0.5300	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0.1700	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0.5200	0
12	0	0	0	0	0	0	0.5300	0
13	0	0	0	0	0	0	0	0
14	0	0	0.4500	0	0	0	0	0
15	0	0	0	0	0	0	0	0

Figure 4.8 Real Value of NaNs in Figure 4.6

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
3	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
4	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
5	0.5000	0.5000	0.6600	0.3600	0.5000	0	0.4900	0.3600
6	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
7	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
8	0.4600	0.4400	0.5200	0.1100	0.5000	0	0.5000	0.2200
9	0.4700	0.3900	0.5000	0.1100	0.5000	0	0.4900	0.4000
10	0.6100	0.6000	0.5500	0.2100	0.5000	0	0.5000	0.2500
11	0.7300	0.6300	0.4200	0.3000	0.5000	0	0.4900	0.2200
12	0.4300	0.5300	0.5200	0.1300	0.5500	0.2200	0.5500	0.2200
13	0.5100	0.5100	0.5200	0.5100	0.5000	0	0.5400	0.2200
14	0.5900	0.4500	0.5300	0.1900	0.5000	0	0.5900	0.2700
15	0.5700	0.4700	0.6000	0.1800	0.5000	0	0.5100	0.2200

Figure 4.9 First Group after KNN Imputation

- **Fourth Step (Imputation):** Impute missing values according to the percentage of missing values in each row. For the rows having more than 20% of missing values, impute MVs using KNN algorithm. For the rows having less than 20% of missing values, impute MVs using SVD. Figure 4.9 shows KNN imputation, where the rows which have more than 20% of missing values are imputed using KNN algorithm based on the shortest distance between missing value and other neighbors. On the other hand, the remaining rows which have less than 20% of missing values are imputed using SVD algorithm based on the updating of the imputed value and then by choosing the best value which are able to achieve the minimum error, as shown in Figure 4.10.

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	0.2853
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	0.5228	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	0.2591	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	0.0084	0.5800	0.3400
9	0.4000	0.3900	0.6000	0.1500	0.5000	0.0079	0.5800	0.3000
10	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
11	0.4200	0.3700	0.5900	0.2000	0.5000	0	0.4679	0.2900
12	0.4000	0.4200	0.5700	0.3500	0.5000	0	0.4800	0.2500
13	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
14	0.6600	0.5500	0.5145	0.1900	0.5000	0	0.4600	0.2200
15	0.4600	0.4400	0.5200	0.1100	0.5000	0	0.5000	0.2200

Figure 4.10 Second Group after SVD Imputation

4.5 Experimental Results

This section presents the analysis of test results obtained using the proposed system framework described previously. The Yeast dataset is used for testing purposes. Missing values are filled using a hybrid method in which KNN deals with rows having more than 20% of missing values and SVD deals with rows which have less than 20% of missing values. For evaluating the performance of the proposed technique, MSE is used to compute the amount of total error existing between the real values and imputed values. The procedure of testing the hybrid method on the Yeast data is done by testing KNN and SVD alone, then later by comparing both methods. To check the quality of the resulted data using KNN or SVD method, the MSE method is used for comparison between the imputation process. Therefore, the first step is using KNN or SVD algorithm alone without activating the threshold value. Threshold value means setting the percentage of missing value in each row. Then, the hybrid method is applied using the same data to compare with the two earlier methods and check the performance of the error value and time respectively.

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	NaN
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	NaN	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	NaN	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	NaN	0.5800	0.3400
9	NaN	0.5000	0.6600	0.3600	0.5000	0	0.4900	NaN
10	0.4000	0.3900	0.6000	0.1500	0.5000	NaN	0.5800	0.3000
11	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
12	0.4200	0.3700	0.5900	0.2000	0.5000	0	NaN	0.2900
13	0.4000	0.4200	0.5700	0.3500	0.5000	0	NaN	0.2500
14	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
15	0.6600	0.5500	NaN	0.1900	0.5000	0	0.4600	0.2200

Fig 4.11 Random Missing Values Generation

It is necessary to use the same generated data to compare because the random generating of NaN values is done by applying random generating function, which means every time the data is different. Figure 4.11 shows missing data generated randomly from the Yeast data and tested using all methods that are mentioned earlier. While Figure 4.12 shows the generated missing data however it replaces NaN with 1 and the other data with 0 respectively. This process in algorithm lets us recognize cell and calculates the percentage for each row.

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	0
9	1	0	0	0	0	0	0	1
10	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0

Figure 4.12 Replace NaN Values in Figure 4.11 with 1

The real value shown in Figure 4.13 must be compared with the imputed data within the suggested algorithm accordingly.

The KNN method applied without using the threshold value for each row is shown in Figure 4.14.This method is effective for the missing value which is more than 20% in each row. But here, this method is run for all cases accordingly.

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0.2200
3	0	0	0	0	0	0	0.5300	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0.7100	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0.5500	0	0	0	0	0	0	0.2200
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0.5200	0
13	0	0	0	0	0	0	0.5300	0
14	0	0	0	0	0	0	0	0
15	0	0	0.4500	0	0	0	0	0

Figure 4.13 Real Value of NaNs in Figure 4.12

Figures 4.14 and 4.15 represent the results of applying the KNN and SVD methods without threshold value which is to be compared later with the other methods.

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	0.2700
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	0.5000	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	0.2200	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	0.3400	0.5800	0.3400
9	0.5000	0.5000	0.6600	0.3600	0.5000	0	0.4900	0.3600
10	0.4000	0.3900	0.6000	0.1500	0.5000	0.3000	0.5800	0.3000
11	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
12	0.4200	0.3700	0.5900	0.2000	0.5000	0	0.5000	0.2900
13	0.4000	0.4200	0.5700	0.3500	0.5000	0	0.5000	0.2500
14	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
15	0.6600	0.5500	0.5000	0.1900	0.5000	0	0.4600	0.2200

Figure 4.14 KNN Imputation without Setting Threshold

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	0.2861
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	0.5241	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	0.2586	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	0.0081	0.5800	0.3400
9	0.5414	0.5000	0.6600	0.3600	0.5000	0	0.4900	0.2947
10	0.4000	0.3900	0.6000	0.1500	0.5000	0.0076	0.5800	0.3000
11	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
12	0.4200	0.3700	0.5900	0.2000	0.5000	0	0.4692	0.2900
13	0.4000	0.4200	0.5700	0.3500	0.5000	0	0.4812	0.2500
14	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
15	0.6600	0.5500	0.5142	0.1900	0.5000	0	0.4600	0.2200

Figure 4.15 SVD Imputation without Setting Threshold

Table 4.1 illustrates the results of the MSE and also running time for both KNN and SVD methods, respectively. It is clear in one's eye that SVD is faster than a fraction of second, but in comparison to it, the error rate is less in the KNN method. However, one should bear in mind the hybrid method is not mentioned here yet.

Table 4.1 MSE Result of KNN and SVD

Algorithm	MSE	Runtime
KNN	0.031	0.16 sec
SVD	0.12	0.14 sec

The hybrid method named as SVDKNN is described below with a threshold condition that the two methods run simultaneously. Therefore, the missing value in each row is checked and compared with the threshold value to decide which method should run for each row.

SVDKNN Algorithm Steps
Input: Testing data (with missing values), Training data (without missing values).
Output: Testing data (with imputed values).

Begin
$Step_1$: Calculate the percentage of missing values for each row in testing data.
$Step_2$: If the percentage of missing values in each row >20%,
impute missing values using SVD
Else
impute missing values using KNN.
$Step_3$: Compute MSE.
End

The resulting of the random data with NaN values are divided into two groups. The first group has more than 20% of the missing data within each row, i.e. there are two or more than two values within the row.

The imputation process follows algorithm SVDKNN, where the SVD method runs first with the threshold of more than 20% then later KNN runs with a threshold of less than 20% respectively. This method is suggested to check all the possibilities that can be applied and checked to see which one is the best. Therefore, SVDKNN algorithm steps are tested first and later in this chapter KNNSVD algorithm steps are checked with the reverse process. Figure 4.16 shows the row with more than 20% missing values.

	1	2	3	4	5	6	7	8
1	NaN	0.5000	0.6600	0.3600	0.5000	0	0.4900	NaN
2	0.4300	0.5300	0.5200	0.1300	NaN	NaN	0.5500	0.2200
3	0.5100	NaN	0.4300	NaN	0.5000	0	0.5200	0.2200
4	NaN	0.4500	NaN	0.1800	0.5000	0	0.5600	0.2600
5	NaN	0.2400	0.3400	0.1600	0.5000	0	0.5000	NaN
6	NaN	0.5300	0.4500	0.1400	0.5000	0	0.5200	NaN
7	NaN	0.4900	0.5300	0.1900	NaN	0	0.5000	0.2200
8	NaN	0.4200	0.5600	0.5500	0.5000	0	NaN	0.2500
9	0.4300	0.5700	NaN	0.4900	0.5000	NaN	0.4800	0.2200
10	0.4600	0.3800	0.4700	0.2200	NaN	0	0.5200	NaN
11	0.6100	0.7200	NaN	0.3300	0.5000	0	0.5800	NaN
12	0.7800	0.7500	0.4000	NaN	0.5000	0	0.5300	NaN
13	NaN	NaN	0.4900	0.4700	0.5000	0	0.4900	0.2700
14	0.4500	0.4500	0.5000	NaN	0.5000	NaN	0.4900	0.2200
15	NaN	0.3400	0.5000	0.1900	NaN	0	0.5000	0.2200

Figure 4.16 First Group with Threshold More than 20%

To check the real data and compare it with the missing NaNs values, Figure 4.17 represents these data.

	1	2	3	4	5	6	7	8
1	0.5500	0	0	0	0	0	0	0.2200
2	0	0	0	0	0.5000	0	0	0
3	0	0.5100	0	0.8700	0	0	0	0
4	0.4000	0	0.5700	0	0	0	0	0
5	0.4200	0	0	0	0	0	0	0.2200
6	0.4100	0	0	0	0	0	0	0.6600
7	0.3800	0	0	0	0.5000	0	0	0
8	0.4900	0	0	0	0	0	0.4200	0
9	0	0	0.5100	0	0	0	0	0
10	0	0	0	0	0.5000	0	0	0.2700
11	0	0	0.3900	0	0	0	0	0.2200
12	0	0	0	0.2800	0	0	0	0.2200
13	0.4000	0.4400	0	0	0	0	0	0
14	0	0	0	0.6000	0	0	0	0
15	0.5100	0	0	0	0.5000	0	0	0

Figure 4.17 Real value of NaNs in Figure 4.16

	1	2	3	4	5	6	7	8
1	0.5479	0.5000	0.6600	0.3600	0.5000	0	0.4900	0.3026
2	0.4300	0.5300	0.5200	0.1300	0.4895	0.0051	0.5500	0.2200
3	0.5100	0.4758	0.4300	0.2531	0.5000	0	0.5200	0.2200
4	0.5033	0.4500	0.4876	0.1800	0.5000	0	0.5600	0.2600
5	0.4266	0.2400	0.3400	0.1600	0.5000	0	0.5000	0.2356
6	0.5000	0.5300	0.4500	0.1400	0.5000	0	0.5200	0.2761
7	0.5031	0.4900	0.5300	0.1900	0.4952	0	0.5000	0.2200
8	0.5337	0.4200	0.5600	0.5500	0.5000	0	0.5276	0.2500
9	0.4300	0.5700	0.5046	0.4900	0.5000	0.0054	0.4800	0.2200
10	0.4600	0.3800	0.4700	0.2200	0.4691	0	0.5200	0.2632
11	0.6100	0.7200	0.5724	0.3300	0.5000	0	0.5800	0.3262
12	0.7800	0.7500	0.4000	0.3041	0.5000	0	0.5300	0.3282
13	0.5246	0.5048	0.4900	0.4700	0.5000	0	0.4900	0.2700
14	0.4500	0.4500	0.5000	0.2473	0.5000	0.0050	0.4900	0.2200
15	0.4708	0.3400	0.5000	0.1900	0.4635	0	0.5000	0.2200

Figure 4.18 SVD Imputation on the First Group

The second group of data that has less than 20% of missing value is presented in Figure 4.19. SVDKNN algorithm steps suggest applying the KNN method in case when the threshold value is set. However, it must be noted that it is better to apply the KNN method for the missing value more than for the 20% of the missing data.

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	NaN
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	NaN	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	NaN	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	NaN	0.5800	0.3400
9	0.4000	0.3900	0.6000	0.1500	0.5000	NaN	0.5800	0.3000
10	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
11	0.4200	0.3700	0.5900	0.2000	0.5000	0	NaN	0.2900
12	0.4000	0.4200	0.5700	0.3500	0.5000	0	NaN	0.2500
13	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
14	0.6600	0.5500	NaN	0.1900	0.5000	0	0.4600	0.2200
15	0.4600	0.4400	0.5200	0.1100	0.5000	0	0.5000	0.2200

Fig 4.19 Second Group with Threshold Less than 20%

In each row, it is better to check all the possibilities and compare the results to present the best method. Here it is necessary and just the right time to mention that if the row is without a missing data, it will be then with less than 20% missing value group. It is much better to show the real values of the NaNs. Figure 4.20 shows exactly these sets of the data respectively.

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0.2200
3	0	0	0	0	0	0	0.5300	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0.1700	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0.5200	0
12	0	0	0	0	0	0	0.5300	0
13	0	0	0	0	0	0	0	0
14	0	0	0.4500	0	0	0	0	0
15	0	0	0	0	0	0	0	0

Figure 4.20 Real Value of NaNs in Figure 4.19

The imputation results after applying the KNN method is shown in Figure 4.21. The criterion of making a comparison between all results for the presented algorithms and methods is MSE method. Table 4.2 shows the error value for applying algorithms KNNSVD and SVDKNN with the threshold value of 20%, 30%, 50%, and 70% MVs respectively. Earlier figures show data of threshold 20% of missing value only, which otherwise will be huge, therefore it is suggested to present this sum of data accordingly.

Table 4.2 MSE Result for Hybrid Techniques (KNNSVD, SVDKNN)

Method	Threshold of Missing Value			
	20%	30%	50%	70%
KNNSVD	0.027	0.029	0.035	0.05
SVDKNN	0.031	0.042	0.049	0.137

	1	2	3	4	5	6	7	8
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800	0.2200
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300	0.2700
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	0.5000	0.2200
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400	0.2200
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800	0.2200
6	0.5100	0.4000	0.5600	0.2200	0.5000	0.5000	0.4900	0.2200
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300	0.2200
8	0.4800	0.4500	0.5900	0.2000	0.5000	0.3400	0.5800	0.3400
9	0.4000	0.3900	0.6000	0.1500	0.5000	0.3000	0.5800	0.3000
10	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300	0.2700
11	0.4200	0.3700	0.5900	0.2000	0.5000	0	0.5000	0.2900
12	0.4000	0.4200	0.5700	0.3500	0.5000	0	0.5000	0.2500
13	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300	0.2200
14	0.6600	0.5500	0.5000	0.1900	0.5000	0	0.4600	0.2200
15	0.4600	0.4400	0.5200	0.1100	0.5000	0	0.5000	0.2200

Figure 4.21 KNN Imputation on the Second Group

KNNSVD Algorithm Steps
Input: Testing data (with missing values), Training data (without missing values).
Output: Testing data (with imputed values).

Begin
$Step_1$: Calculate the percentage of missing values for each row in testing data.
$Step_2$: If the percentage of missing values in each row >20%,
impute missing values using KNN
Else
Impute missing values using SVD.
$Step_3$: Compute MSE.
End

The hybrid approach, apparently, is derived by combining both KNN and SVD correlation of the data matrix. Hence, by this, using the hybrid method may achieve higher imputation than a single approach only.

By comparing the results of MSE for KNNSVD and SVDKNN with different thresholds value as in Table 4.2, We can recognize the KNNSVD and SVDKNN achieve the best of results whenever the threshold is 20% as Figure 4.23 shows.

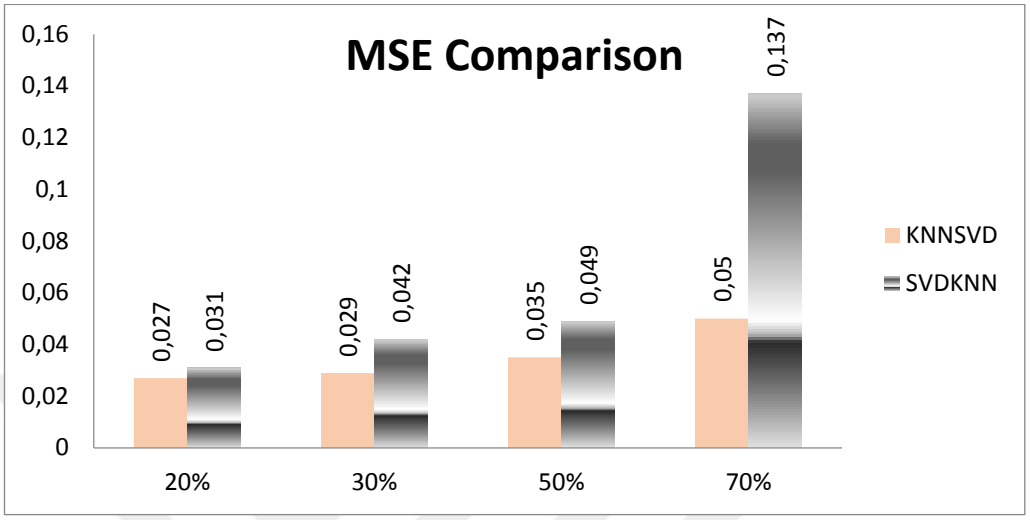


Fig 4.23 MSE Result Comparison between KNNSVD and SVDKNN

Tables 4.3 indicate the result of comparison between the four tested methods which are KNN alone, SVD alone, best of KNNSVD (when the threshold is 20%), best of SVDKNN (when the threshold is 20%).

Table 4.3 MSE Results Comparison for the Tested Methods

Method	MSE	TIME
KNN	0.031	0.16 sec
SVD	0.12	0.14 sec
KNNSVD	0.027	0.54 sec
SVDKNN	0.031	0.39 sec

According to the results that are shown in Table 4.3 and the Figure 4.22 we can conclude, that the KNNSVD method achieve the minimum error 0.027 with a time of 0.54 sec on the other hand, SVD alone that has the maximum error with a value of 0.12 and a time of 0.14 sec.

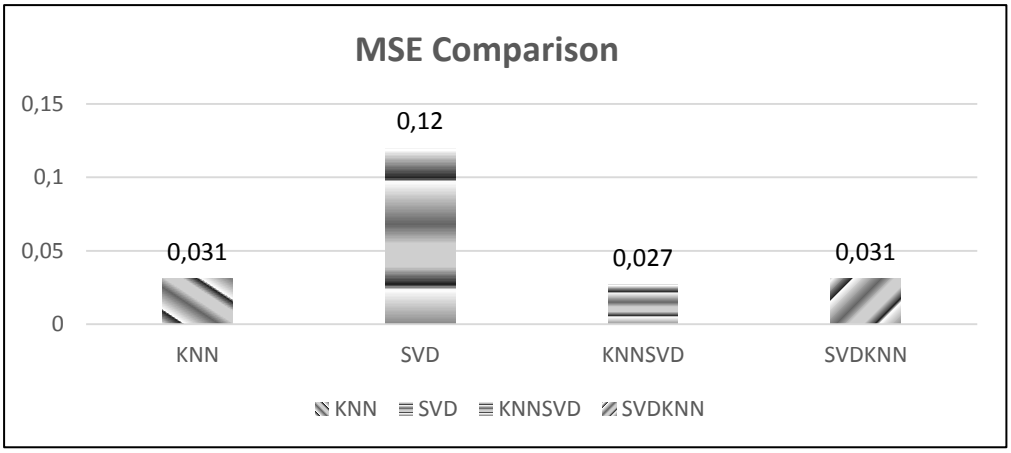


Fig 4.22 MSE Result Comparison between KNN, SVD, KNNSVD, SVDKNN

Finally the MSE result of KNNSVD method is compared with different studies for literature as shown in Figure 4.24 shows.

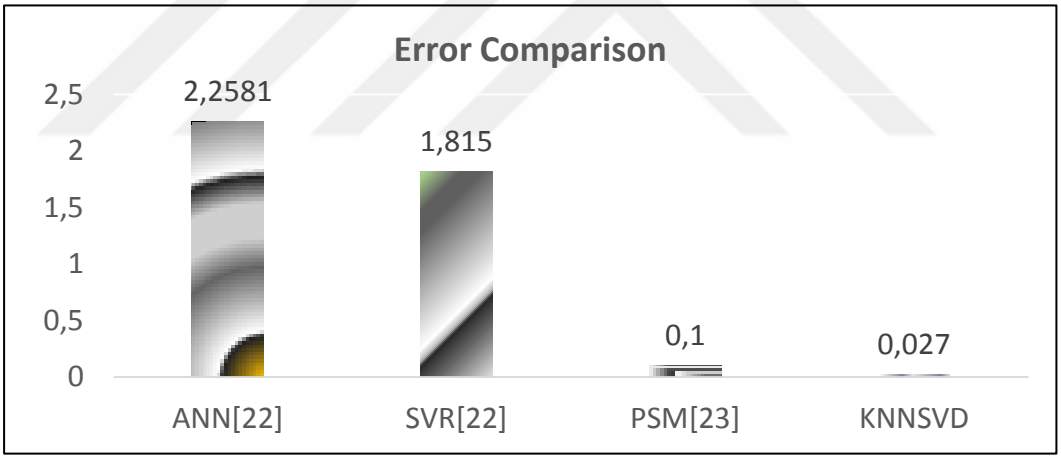


Fig 4.24 Error Comparison between KNNSVD and other Previous Methods

According the results that are shown in Figure 4.24, the ANN method has the maximum value of error followed by SVR with a value of 1.815 and PSM with a value of 0.1. The proposed method has 0.027 error rate. Since the given tests were performed using different datasets, normally they are not directly comparable. However, this comparison is given to provide an idea about the proposed method.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The conclusions that can be summarized from the proposed system are as follows:

- For preparing any data for data mining operations like classification and correlation, it is extremely necessary handle the missing value problem.

- The suggested hybrid method in this thesis imputes all records having more than 20% NaN values using KNN and all records having less than 20% NaN values using SVD.

- According to the results of Table 4.1, the KNN algorithm imputes missing values with the lowest error than the SVD algorithm when used alone on the whole Yeast data without using any threshold.

- According to the results in Tables 4.2, MSE value increases whenever the threshold of missing values are increased.

- Experimental results show that the efficiency of KNNSVD technique is better than SVDKNN technique that achieves 0.027, 0.029, 0.035, 0.05 when the thresholds of missing values are 20%, 30%, 50%, and 70% in contrast to SVDKNN that achieved 0.031, 0.042, 0.49, 0.137 when the threshold of missing values are 20%, 30%, 50%, and 70%, respectively.

- By comparing the results of the proposed hybrid method with the previous results given in Chapter 2, we can demonstrate that the proposed method

presents an efficient result in missing data imputation better than the previous methods.

5.2 Recommendations for Future Work

The work in this thesis can be updated in some ways by enhancing the suggested procedure as mentioned below:

- The proposed method can be applied on different data sets such as healthcare, medical, crime, police, etc.

- Different data mining algorithms such as support vector machine, artificial neural network and random forest can be applied for imputation missing values.

References

[1] Purwar, A., & Singh, S. K. (2014, December). Issues in data mining: A comprehensive survey. In Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on (pp. 1-6), IEEE.

[2] Rubin, D. B. (1987). Multiple imputations for nonresponse in surveys (Wiley series in probability and statistics).

[3] Garciarena, U., & Santana, R. (2017). An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. Expert Systems with Applications, 89, 52-65.

[4] Baraldi, A. N., & Enders, C. K. (2010). An introduction to modern missing data analyses. Journal of school psychology, 48(1), 5-37.

[5] Eekhout, I., de, V. H. C. W., Twisk, J. W. R., Brand, J. P. L., de, B. M. R., & Heymans, M. W. (March 01, 2014). Missing data in a multi-item instrument were best handled by multiple imputation at the item score level. Journal of Clinical Epidemiology, 67, 3, 335-342.

[6] Enders, C. K. (2011). Missing not at random models for latent growth curve analyses. Psychological methods, 16 (1), 1.

[7] Zhang, S. (2012). Nearest neighbor selection for iteratively kNN imputation. Journal of Systems and Software, 85(11), 2541-2552.

[8] Aydilek, I. B., & Arslan, A. (2013). A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm. Information Sciences, 233, 25-35.

[9] Arciniegas-Alarcón, S., García-Peña, M., Krzanowski, W., & dos Santos Dias, C. T. (2014). Imputing missing values in multi-environment trials using the singular value decomposition: An empirical comparison. Communications in Biometry & Crop Science, 9(2).

[10] Tutz, G., & Ramzan, S. (2015). Improved methods for the imputation of missing data by nearest neighbor methods. Computational Statistics & Data Analysis, 90, 84-99.

[11] Peng, C. Y. J., Harwell, M., Liou, S. M., & Ehman, L. H. (2006). Advances in missing data methods and implications for educational research. Real data analysis, 3178.

[12] Medhanie, A. G. (2013). The robustness of multilevel multiple imputation for handling missing data in hierarchical linear models.

[13] Sinharay, S., Stern, H. S., & Russell, D. (2001). The use of multiple imputation for the analysis of missing data. Psychological methods, 6(4), 317.

[14] Horton, N. J., & Kleinman, K. P. (2007). Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models. The American Statistician, 61(1), 79-90.

[15] Stuart, E. A. (2010, June). Recent advances in missing data methods: multiple imputation by chained equations. In Health Annual Research Meeting.

[16] Eekhout, I., De Boer, M., Twisk, J., De Vet, H., & Heymans, M. (2012). Brief Report: Missing Data: A Systematic Review of How They Are Reported and Handled. Epidemiology, 23(5), 729-732. Retrieved from http://www.jstor.org/stable/41739653.

[17] Zhang, Z. (2016). Missing data imputation: focusing on single imputation. Annals of translational medicine, 4(1).

[18] Baraldi, A. N., & Enders, C. K. (2010). An introduction to modern missing data analyses. Journal of school psychology, 48(1), 5-37.

[19] Schafer, J. L. (1999). Multiple imputation: a primer. Statistical methods in medical research, 8(1), 3-15.

[20] Graham, J. W. (2009). Missing data analysis: Making it work in the real world. Annual review of psychology, 60, 549-576.

[21] Schafer, J. L. (1997). Analysis of incomplete multivariate data. Chapman and Hall/CRC.

[22] Richman, M. B., Trafalis, T. B., & Adrianto, I. 1School of Meteorology, 2 School of Industrial Engineering, University of Oklahoma, Norman, OK.

[23] Paul, A., & Sil, J. (2011, June). Estimating missing value in microarray gene expression data using fuzzy similarity measure. In Fuzzy Systems (FUZZ), 2011 IEEE International Conference on (pp. 1890-1895). IEEE.

[24] Susianto, Y., Notodiputro, K. A., Kurnia, A., & Wijayanto, H. (2017, March). A Comparative Study of Imputation Methods for Estimation of Missing Values of Per Capita Expenditure in Central Java. In IOP Conference Series: Earth and Environmental Science (Vol. 58, No. 1, p. 012017). IOP Publishing.

[25] Anchalia, P. P., & Roy, K. (2014, January). The k-nearest neighbor algorithm using MapReduce paradigm. In Intelligent Systems, Modelling and Simulation (ISMS), 2014 5th International Conference on (pp. 513-518). IEEE.

[26] Luengo, J., García, S., & Herrera, F. (2012). On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowledge and information systems, 32(1), 77-108.

[27] Hruschka, E. R., Hruschka Jr, E. R., & Ebecken, N. F. (2003). A Nearest-Neighbor Method as a Data Preparation Tool for a Clustering Genetic Algorithm. In SBBD (pp. 319-327).

[28] de Silva, H. M., & Perera, A. S. (2017). Evolutionary k-nearest neighbor imputation algorithm for gene expression data. ICTer, 10(1).

[29] Arciniegas-Alarcón, S., García-Peña, M., Krzanowski, W., & dos Santos Dias, C. T. (2014). Imputing missing values in multi-environment trials using the singular value decomposition: An empirical comparison. Communications in Biometry & Crop Science, 9(2).

[30] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., ... & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6), 520-525.

APPENDICES

APPENDIX A- General Terms

Missing Value: it is value that is not stored for a variable in the observations.
İmputation Missing Values: it is process of replacing missing data with substituted values.

Data mechanisms: it is based on the relationship between the missing data and observed values.

Hybrid imputation: it combines two methods in hybrid model that is able to impute missing data.

Mean Square Error (MSE): it is the average square difference between the estimated value and what is estimated.

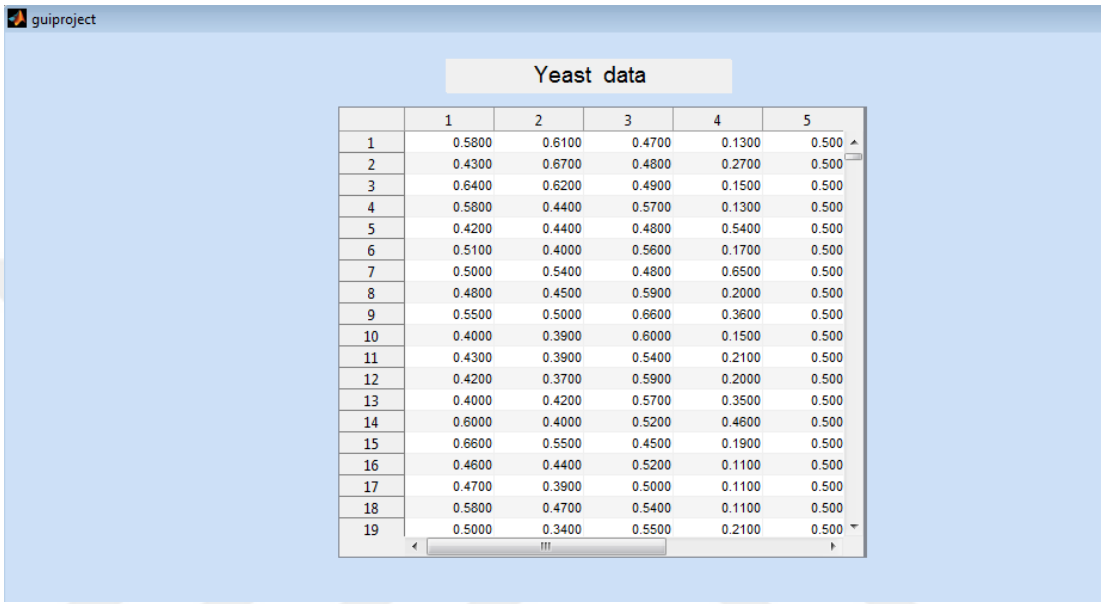
Missing Values Random Generation: it is process that generate patterns of missing data in random model. This operation is useful when simulating or testing algorithms.

Euclidean distance: it is the ordinary straight-line distance between two points in Euclidean space.

Normalize Root Mean Square Error (NRMSE): is frequently used measure of differences between values predicted by a model or an estimator and values observed.

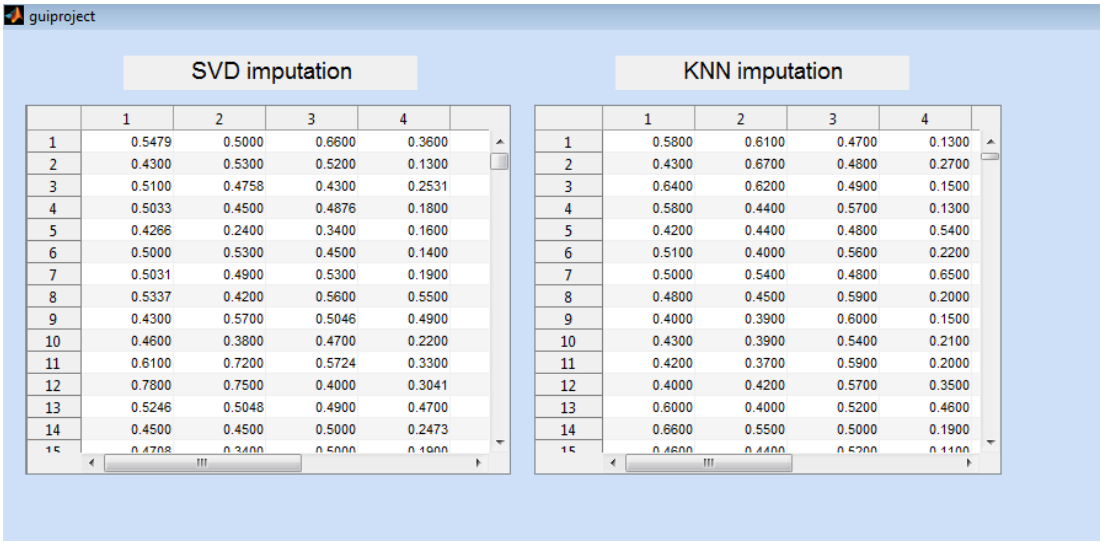
APPENDIX B- Implementation Screenshots

This appendix shows some screenshots of imputation methods when they are in hybrid model and alone model.



	1	2	3	4	5
1	0.5800	0.6100	0.4700	0.1300	0.500
2	0.4300	0.6700	0.4800	0.2700	0.500
3	0.6400	0.6200	0.4900	0.1500	0.500
4	0.5800	0.4400	0.5700	0.1300	0.500
5	0.4200	0.4400	0.4800	0.5400	0.500
6	0.5100	0.4000	0.5600	0.1700	0.500
7	0.5000	0.5400	0.4800	0.6500	0.500
8	0.4800	0.4500	0.5900	0.2000	0.500
9	0.5500	0.5000	0.6600	0.3600	0.500
10	0.4000	0.3900	0.6000	0.1500	0.500
11	0.4300	0.3900	0.5400	0.2100	0.500
12	0.4200	0.3700	0.5900	0.2000	0.500
13	0.4000	0.4200	0.5700	0.3500	0.500
14	0.6000	0.4000	0.5200	0.4600	0.500
15	0.6600	0.5500	0.4500	0.1900	0.500
16	0.4600	0.4400	0.5200	0.1100	0.500
17	0.4700	0.3900	0.5000	0.1100	0.500
18	0.5800	0.4700	0.5400	0.1100	0.500
19	0.5000	0.3400	0.5500	0.2100	0.500

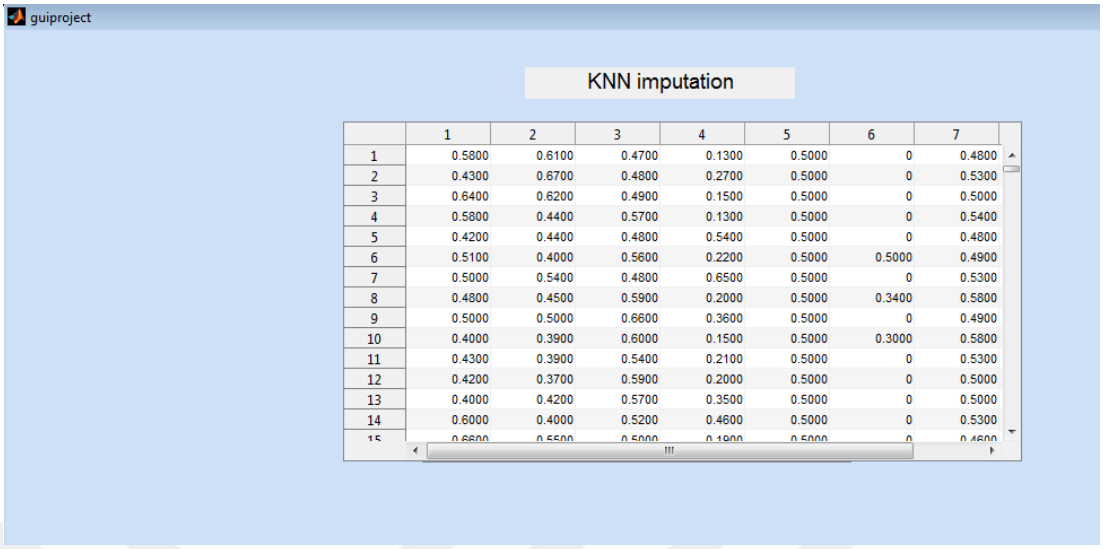
Screenshot of Yeast



	1	2	3	4
1	0.5479	0.5000	0.6600	0.3600
2	0.4300	0.5300	0.5200	0.1300
3	0.5100	0.4758	0.4300	0.2531
4	0.5033	0.4500	0.4876	0.1800
5	0.4266	0.2400	0.3400	0.1600
6	0.5000	0.5300	0.4500	0.1400
7	0.5031	0.4900	0.5300	0.1900
8	0.5337	0.4200	0.5600	0.5500
9	0.4300	0.5700	0.5046	0.4900
10	0.4600	0.3800	0.4700	0.2200
11	0.6100	0.7200	0.5724	0.3300
12	0.7800	0.7500	0.4000	0.3041
13	0.5246	0.5048	0.4900	0.4700
14	0.4500	0.4500	0.5000	0.2473
15	0.4708	0.3400	0.5000	0.1900

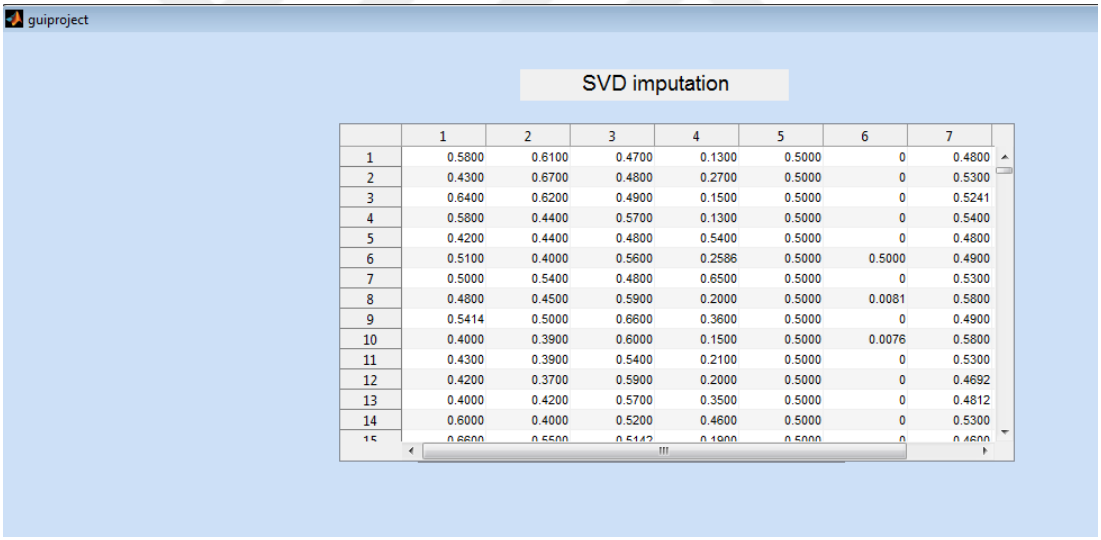
	1	2	3	4
1	0.5800	0.6100	0.4700	0.1300
2	0.4300	0.6700	0.4800	0.2700
3	0.6400	0.6200	0.4900	0.1500
4	0.5800	0.4400	0.5700	0.1300
5	0.4200	0.4400	0.4800	0.5400
6	0.5100	0.4000	0.5600	0.2200
7	0.5000	0.5400	0.4800	0.6500
8	0.4800	0.4500	0.5900	0.2000
9	0.4000	0.3900	0.6000	0.1500
10	0.4300	0.3900	0.5400	0.2100
11	0.4200	0.3700	0.5900	0.2000
12	0.4000	0.4200	0.5700	0.3500
13	0.6000	0.4000	0.5200	0.4600
14	0.6600	0.5500	0.5000	0.1900
15	0.4600	0.4400	0.5200	0.1100

Screenshot of SVDKNN hybrid imputation model



	1	2	3	4	5	6	7
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	0.5000
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800
6	0.5100	0.4000	0.5600	0.2200	0.5000	0.5000	0.4900
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300
8	0.4800	0.4500	0.5900	0.2000	0.5000	0.3400	0.5800
9	0.5000	0.5000	0.6600	0.3600	0.5000	0	0.4900
10	0.4000	0.3900	0.6000	0.1500	0.5000	0.3000	0.5800
11	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300
12	0.4200	0.3700	0.5900	0.2000	0.5000	0	0.5000
13	0.4000	0.4200	0.5700	0.3500	0.5000	0	0.5000
14	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300
15	0.6600	0.5500	0.5000	0.1900	0.5000	0	0.4600

Screenshot of KNN alone imputation method



	1	2	3	4	5	6	7
1	0.5800	0.6100	0.4700	0.1300	0.5000	0	0.4800
2	0.4300	0.6700	0.4800	0.2700	0.5000	0	0.5300
3	0.6400	0.6200	0.4900	0.1500	0.5000	0	0.5241
4	0.5800	0.4400	0.5700	0.1300	0.5000	0	0.5400
5	0.4200	0.4400	0.4800	0.5400	0.5000	0	0.4800
6	0.5100	0.4000	0.5600	0.2586	0.5000	0.5000	0.4900
7	0.5000	0.5400	0.4800	0.6500	0.5000	0	0.5300
8	0.4800	0.4500	0.5900	0.2000	0.5000	0.0081	0.5800
9	0.5414	0.5000	0.6600	0.3600	0.5000	0	0.4900
10	0.4000	0.3900	0.6000	0.1500	0.5000	0.0076	0.5800
11	0.4300	0.3900	0.5400	0.2100	0.5000	0	0.5300
12	0.4200	0.3700	0.5900	0.2000	0.5000	0	0.4692
13	0.4000	0.4200	0.5700	0.3500	0.5000	0	0.4812
14	0.6000	0.4000	0.5200	0.4600	0.5000	0	0.5300
15	0.6600	0.5500	0.5142	0.1900	0.5000	0	0.4600

Screenshot of SVD alone imputation method