Evolutionary k-Nearest Neighbor Imputation Algorithm for Gene Expression Data

Hiroshi de Silva, A. Shehan Perera

***Abstract*—Large data sets are produced by the gene expression process which is done by using the DNA microarray technology. These gene expression data are recognized as a common data source which contains missing expression values. In this paper, we present a genetic algorithm optimized k- Nearest neighbor algorithm (Evolutionary kNNImputation) for missing data imputation. Despite the common imputation methods this paper addresses the effectiveness of using supervised learning algorithms for missing data imputation. Missing data imputation approaches can be categorized into four main categories and among the four approaches, our focus is mainly on local approach where the proposed Evolutionary k- Nearest Neighbor Imputation Algorithm falls in. The Evolutionary k- Nearest Neighbor Imputation Algorithm is an extension of the common k-nearest Neighbor Imputation Algorithm which the genetic algorithm is used to optimize some parameters of k- Nearest Neighbor Algorithm. The selection of similarity matrix and the selection of the parameter value *k* can be identified as the optimization problem. We have compared the proposed Evolutionary k- Nearest Neighbor Imputation algorithm with k-Nearest Neighbor Imputation algorithm and mean imputation method. The three algorithms were tested using gene expression datasets. Certain percentages of values are randomly deleted in the datasets and recovered the missing values using the three algorithms.**

Results show that Evolutionary kNNImputation outperforms kNNImputation and mean imputation while showing the importance of using a supervised learning algorithm in missing data estimation. Even though mean imputation happened to show low mean error for a very few missing rates, supervised learning algorithms became effective when it comes to higher missing rates in datasets which is the most common situation among datasets.

***Keywords*— Missing data imputation,kNNImputation, EvlkNNImputation, Genetic algorithm optimization, Supervised learning algorithm, Big data, Similarity metric, Gene expression data, Evolutionary algorithms**

Manuscript received on 14th November, 2016. Recommended by Dr. T.M.H.A.Usoof on 4th May, 2017. This paper is an extended version of the paper, " Missing Data Imputation using Evolutionary k- Nearest Neighbor Algorithm for Gene Expression Data" presented on ICTer2016 Conference.

Hiroshi de Silva is an instructor and research student at the Department of Computer Science and Engineering in the field of data mining and machine learning. (e-mail: hiroshides@cse.mrt.ac.lk)

Dr. Shehan Perera is a senior Lecturer at the Department of Computer Science and Engineering in the fields of database, big data, machine learning and software engineering. (e-mail: shehan@cse.mrt.ac.lk)

I. Introduction

DNA microarray technology is widely used to analyze gene expression data. These expression data sets are large and frequently found with some missing values. Missing values of gene expression data occur for many reasons such as: insufficient resolution, image corruption, due to dust or scratches on the slide, or as a result of the robotic methods used to create them [1]. In gene expression analysis missing values rate of less than 1% is considered inconsequential, 1-5% is controllable, 5-15% requires refined methods to handle the imputation, and more than 15% strictly influences the prediction or interpretation [2]. Given the expense of collecting data, we cannot afford to start over or to wait until wedevelop fool proof methods of gathering information[3]. As it is very time consuming and expensive to repeat the process, scientists are now moving into missing data imputation as a solution [4]. In this paper, we present a genetic algorithm optimized k-nearest neighbor algorithm imputing missing data compared to the k- Nearest Neighbor Imputation Algorithm and several other common imputation methods. The importance of using a machine learning algorithm is discussed in this paper as most of the common imputation methods such as: case deletion and mean imputation method are showing less effective results by not considering the correlation of data.

A. Gene Expression Data

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product[5]. The genes encode proteins and proteins are responsible for dictating cell functions. Therefore, the final outcome of the gene expression process is usually a protein which is a gene product. Gene expression process consists of 2 main steps: Transcription and Translation. Transcription is when the DNA in a gene is copied to produce an RNA a messenger RNA (mRNA) is being called. This mRNA is carried out by an enzyme called RNA polymerase which uses available bases from the nucleus of the cell to form the mRNA.RNA is chemically similar in structure and properties to DNA, but it only has a single strand of bases. Instead of the base thymine (T), RNA has a base called uracil (U).

Translation occurs after the mRNA has carried the transcribed ‘message’ from the DNA to protein-making factories in the cell, called ribosomes. The message carried by the mRNA is read by a carrier molecule called transfer RNA (tRNA).The mRNA is reading three letters (a codon) at a time.

Each codon specifies a particular amino acid. For example, the three bases 'GGU' code for an amino acid called glycine. As there are only 20 amino acids but 64 potential combinations of codon, more than one codon can code for the same amino acid. For example, the codons 'GGU' and 'GGC' both code for glycine. Each amino acid is attached specifically to its own tRNA molecule. When the mRNA sequence is read, each tRNA molecule delivers its amino acid to the ribosome and binds temporarily to the corresponding codon on the mRNA molecule. Once the tRNA is bound, it releases its amino acid and the adjacent amino acids all join together into a long chain called a polypeptide. This process continues until a protein is formed.The data formed as a result of the genes expression process is really important in identifying mutations and alterations in genes.

The DNA microarray technology is used to monitor expression data under variety of conditions [6]. Scientists are using microarray technology to study biological processes of gene expression data in human tumors to yeast sporulation [7], [8]. Also, with the microarray technology to generate gene expression data some spots on the array may be missing due to various factors (for example, machine error) [4].

B. Missing Data Imputation Methods

Generally, most methods handle missing data simply by discarding missing data but discard of missing data can lead to estimates with larger standard errors due to reduced sample size [9]. Dropping such cases with missing data has yield biased or inconclusive results even though such techniques are still widely used in software engineering [10]. This method is known as "complete case analysis". Another method of ignoring missing data from datasets is the "available case analysis" where different subsets of data are taken to different aspects of the same study due to inability in taking the full dataset because some values of variables have incomplete data. This approach excludes some variables which are needed to satisfy the assumptions necessary for desired interpretations [9]. Complete case analysis and available case analysis both are reducing the sample sizes of the datasets. Missing value imputation of numerical data is mostly handled in general by mean substitution in several works [10]. The main disadvantage is that this method can distort the distribution for the variable which is used for imputation by underestimating the standard deviation [9]. Median imputation is also used to assure robustness since mean is affected by outliers of adataset. For the categorical attributes, the mode imputation is used instead of mean and median of a dataset [11] .

The disadvantage of this method is that it does not consider dependencies among attribute values [12]. Another widely used method is multiple imputation which missing values are redicted using existing values from other variables. This process is performed multiple times, producing multiple imputed data sets[13] .

Missing value estimation approaches can be categorized into four main categories such as: Global approach, Local approach, Hybrid approach, and Knowledge based approach.

1) Global Approach: In global approach, algorithm does the missing value estimation by looking into the entire data matrix with global correlation information. According to [14], if the algorithms assume that there exists a global covariancestructure in all genes samples and that the genes exhibitdominant local similarity structures, then the imputation willbecome less accurate. Examples of algorithms which use the global approach are SVDImputation [1] and Bayesian Principal Component Analysis (BPCA) [15].

2) Local Approach: In local approach, algorithms take the local similarity structures of data matrix in order to do the missing value estimation. The subsets of genes that show high correlation with the genes that contain the missing values are used to compute the missing values. The KNN imputation (KNNimpute) and local least square imputation (LLSimpute) are some of the common and efficient algorithms for local approach. The KNNImpute[1] takes the pairwise informationbetween the target gene with missing values and the reference genes to do the imputation of missing data. The LLSImpute [16] uses a multiple regression model to impute missing values.Sequential LLSimpute (SLLSimpute) [17]is an extension of LLSimpute algorithm which performs imputation sequentially by starting from the gene with least missing rates. The imputed genes are then reused for imputation of other genes. It has been proven that SLLSimpute performs better than LLSimpute because the genes with missing values are reusable in this algorithm.

3) Hybrid Approach: Heterogeneous data sets require a local approach as local correlation beween genes are used to do the missing value estimation. There are some data sets which require global approaches because of the global correlation structure of data. There are some hybrid methods like LinCmb [18] that can capture both local and global correlation information in the data sets. Using this method, the missing values are estimated by convex combination of five different imputation methods such as: row average, KNNimpute, SVDimpute, BPCA, and GMCimpute. The LinCmb generates fake missing entries where the true values are known and uses the constituent methods to estimate the missing entries. LinCmb is also adaptive to the correlation structure of data matrix where more missing entries are present, global methods will become the focus to determine the missing values.

4) Knowledge Assisted Approach: This approach integrates the domain knowledge or external information into the missing values imputation process. This is a powerful approach as it significantly improves the accuracy of imputation using domain knowledge. Also, this approach performs well on data sets with small number of samples which are noisy or have high missing rate. An example of this approach is the

Projection Onto Convex Set (POCS), which exploits the biological occurrence of synchronization loss and correlation information between genes. Histone Acetylation Information Aided Imputation (HAIimpute) is another example which combines histone acetylation information into KNNimpute and LLSimpute to improve the accuracy of missing value estimation [19].

Many machine learning algorithms solve missing data problem in an efficient way. One advantage of using a machine learning approach is that the missing data treatment is independent of the learning algorithm used [20]. Most common algorithms used in imputation are EM algorithm [21], SVDImpute[1], CN2 Induction Algorithm [22, p. 2], C4.5 Algorithm, the local least squares imputation method (LLS) [16], the Bayesian principal component analysis (BPCA) [15]and kNNImpute[20], [11], [23]. Commonly used imputation methods for gene expression data use clustering technique [24]-[25] and techniques based on supervised learning [26], [27]. Also, it has been found that some approaches are not compatible for missing data imputation because missing values are causing negative effects on support vector machines (SVM), single value decomposition (SVD), and principal component analysis (PCA) as these methods cannot function on data with missing values [14].

C. Genetic Algorithm Optimization

Genetic algorithms have been widely used for many optimization problems. Genetic algorithms are used for feature selection and optimization of many algorithms [28], [29]. Genetic algorithms are widely used for many optimization needs. The genetic algorithm searches for the optimal solution. The process of the Genetic Algorithm is illustrated in Fig. 1. First, it creates a population of strings. These strings are named as chromosomes. In general, these chromosomes are represented by a bit string (a binary string with 1 s and 0s). This population is a collection of candidate solutions for a defined problem. A single solution in the population is referred to as an individual. A fitness score is calculated for each individual to measure how "good" the individual is which represent a solution. The highest the fitness value is, better the solution. Two individuals which are more fit are selected out of this population. This selection process is based on the concept of "Survival of the fittest" in the natural world. This fitness function is used to measure the quality of an individual in order to increase its probability of survival throughout the evolutionary process. After that, these individuals are selected for "breeding" where they reproduce another two new individuals (offspring). This is done by the crossover operator in genetic algorithms. Crossover creates two offspring strings which are copied from the parent strings with highest fitness value. During each new generation of individuals, there is a chance for each individual to mutate. There is a random chance for individuals to get change in a small way than their parents by changing the value of a single bit in the string. This process will continue until the algorithm meets its termination conditions. Each iteration of selecting the most fit individuals, cross-over and mutation is called a generation.

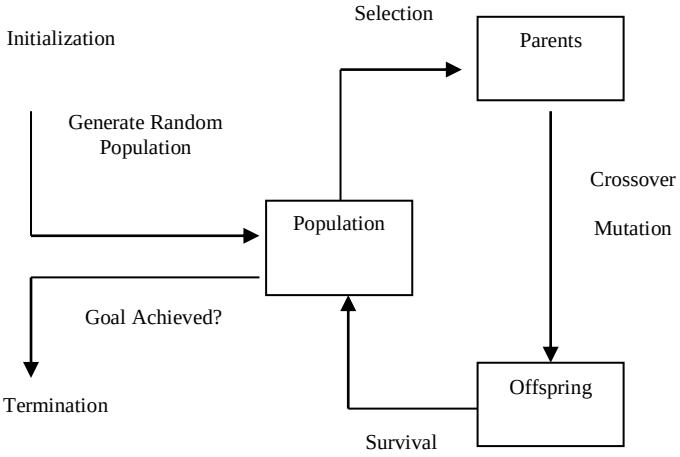

Fig. 1. Genetic Algorithm

There is not a single way in the termination condition mentioned above because there are many ways to end the algorithm. One way is to run the search for a number of generations. The longer the algorithm runs the better. Another approach is to end the algorithm after a certain number of generations pass with no improvement in the fitness of the best individual in the population [30]. The simplest genetic algorithm uses fitness-proportionate selection, single-point crossover and single-point mutation.

Also, the genetic algorithms are commonly used with k-nearest neighbor algorithm as k-NN mainly deals with larger datasets. The k-NN algorithm with genetic algorithms can be used to get weighted vectors for attributes in a dataset[31],[32]. We have used the genetic algorithm to optimize k-NN algorithm before it performs the imputation. The genetic algorithm assigns weights to each attribute and finds weight vector for attributes. Also the genetic algorithm is used to find the optimum value for the parameter k which indicates the number of neighbors to look up in majority vote. The system architecture which illustratesthe methodology of genetic algorithm optimized kNNImpute system is illustrated below in Fig. 2.

II. METHODOLOGY

In this paper, we are presenting a genetic algorithm optimized k- nearest neighbor algorithm for missing data imputation namely EvlkNNImpute. The kNNImpute has been showing successful results compared to many other approaches used in imputation [1],[20]. The advantages of k-NN Imputation are it does not require to create a predictive model for each attribute with missing values in the dataset, treat instances with multiple missing values, it considers the correlation structure of data, and predict both qualitative and quantitative attributes [11]. The disadvantages are the results depend on the parameter *k*and the time required by the algorithm to calculate the distance between instances. Therefore, larger the dataset more

timeis required by the algorithmto do the imputation. In order to overcome those disadvantages in kNNImpute, we introduce EvlkNNImpute which runs an optimization method using the genetic algorithm. This genetic algorithm optimization will result the optimized *k* for a given dataset and assign weights to each attribute/ feature in a dataset.As in Figure 1, the algorithmneeds to be trained using a training dataset as the initial step. The training dataset should not have missing values because the evolutionary k- nearest neighbor is implemented to run an optimization process before the imputation where weight values will be assigned to each attribute of the dataset based on the importance of the attributes towards the prediction of missing value.

Therefore, data with missing values can be separated to another dataset and leave the instances with complete values as the training dataset. The separated data instances with missing values will not be taken to produce the trained model by the genetic algorithm as the missing values in certain attributes will create bias when assigning weights. As an example, the hair color of a person is more important when predicting whether a person is an Asian or not rather than the attributes like weight and height. Once the training dataset is determined, the genetic algorithm will assign weights to each attribute. While the genetic algorithm runs several iterations/ evolutions, a set of weights for all attributes will be given at each iteration.

A fitness score will also be calculated for each iteration using the genetic algorithm and that is to indicate how good the solution is. Higher the fitness score, better the solution. Table II provides an example set of weights assigned to each attribute of a dataset at the highest fitness score of a dataset. The optimum *k* given for the dataset will also be the value of the parameter k at the highest fitness score.

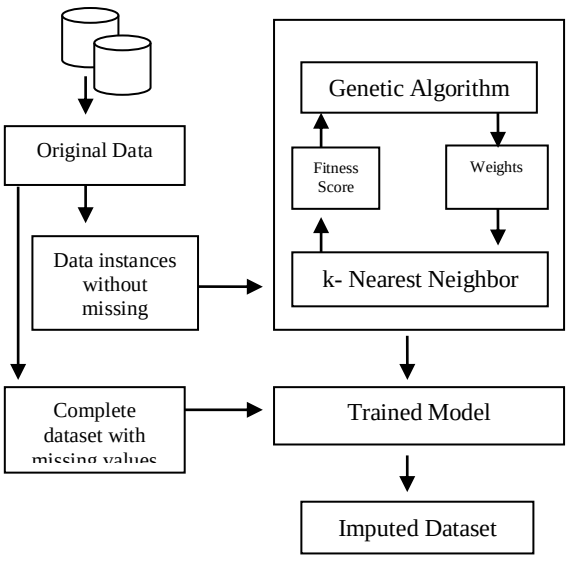

Fig.2.Genetic algorithm optimized k- nearest neighbor algorithm

The fitness function is defined for this application is as below:

$$Fit = \left[1/\sqrt{\sum_{0}^{m-1}(1-Conf)^2}\right] \text{x } 100$$

The above fitness score is computed assuming an appropriate classification confidence measure for the respective application. Usually, a genetic algorithm runs until it meets a threshold or until there is no improvement in the fitness score [31][33]. In this application we have used the latter where several iterations of genetic algorithm run until there is no improvement in the fitness score. The weights set with the highest fitness score is saved as a model to be used during the imputation process where missing values get estimated. The selection of a similarity metric to identify the neighbors and the selection of the optimum k as the number of neighbors can be considered as the optimization problem where we use the genetic algorithm for.

The similarity functions are defined as follows:

Nominal attributes, $di = \begin{cases} W_i & X_i = Y_i \\ 0 & X_i \neq Y_i \end{cases}$

Numeric attributes, $d_i = W_i (X_i - Y_i)$

Similarity, $Sim = \frac{1}{\sum_{i=0}^{n-1} d_i}$

Upon the completion of training the algorithm, the weights given in the saved model during the training process and the optimum k given by the algorithm will be used in predicting missing values of the dataset. If a value in a certain data column/ attribute is missing, the algorithm will look for the weight assigned for the attribute in the saved model and the optimum k assigned by the genetic algorithm. Based on the optimum number of neighbors (k), relevant similarity metric and the weights the missing valueswill be replaced by the algorithm. Likewise, all missing values will get imputed by the Evolutionary k- nearest neighbor algorithm at the end.

III. Evaluation and Results

We compare results using three imputation methods. The mean imputation method, kNNImputation method and proposed EvlkNNImputation method. The comparison of three imputation methods was done using three different yeast datasets which are illustrated in Table I. The “seq” dataset used in the evaluation is not a dataset with sequences in it. The dataset consists of attributes related to sequences. The attributes of “seq” dataset are mentioned in TABLE II. To assess the performance of missing value estimation methods, we randomly deleted values in “seq” dataset and two microarray datasets named “gasch2” [34] and “spo” [8] with certain missing rates. The datasets were taken from the study “Predicting gene function in Saccharomyces cerevisiae” [34] . Data are removed from the original data sets in order to

produce artificially incomplete data sets for imputation and to have the total control over the missing data in the dataset. Reference [20] states if some test set has missing data, then the inducer's ability to classify missing data properly may influence on the result and that influence is undesirable since the objective of this type of work is to analyze the viability of the imputation method.

TABLE I
DESCRIPTION OF DATASETS USED

Dataset	Description	Features	Instances	Time taken for optimization
gasch2	Microarray data	51	204	28.85sec
spo	Microarray data, sporulation in budding yeast	76	1597	19.8 min
seq	Attributes calculated from sequence alone	14	500	44.12 sec

Fig. 3 and Fig. 4 show the genetic algorithm optimization results of EvlkNNImpute algorithm for two datasets. The x-axis indicates the evolutions or iterations of the genetic algorithm. The y- axis indicates the fitness score of each evaluation/ iteration. The genetic algorithm runs until certain number of generations pass with no improvement in the fitness of the best individual in the population [30].

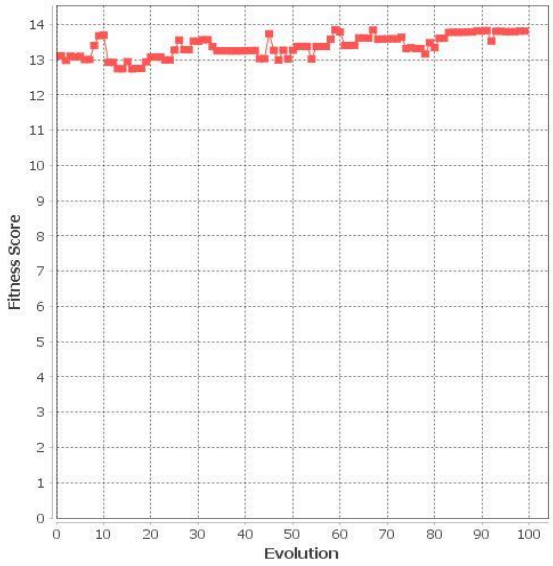

Fig.3. Fitness score over iteration/ evolution for the gasch2 dataset when imputing the missing values of the attribute which indicates the 5 minutes heat shock effect with optimum k value of 3.

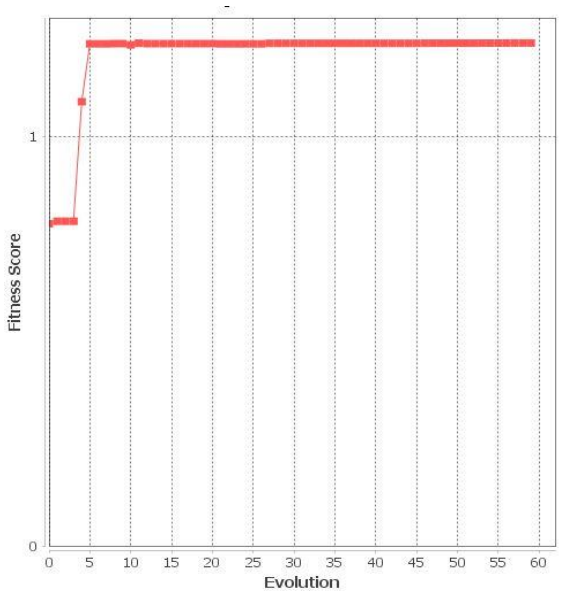

Fig.4.Fitness score over iteration/evolution for the seq datasetwhen imputing the missing values of the attribute molecular weight with optimum k value of 10.

The time taken for genetic algorithm optimization is increasing with the number of instances or the attributes as illustrated in Table I. When the numbers of attributes get increased the algorithm takes more time for optimization than when the numbers of instances get increase. The weights assigned to each feature/attribute by the Evolutionary k-NN for the sequence dataset is shown in Table II.

TABLE II
WEIGHTS ASSIGNED TO ATTRIBUTES IN SEQ DATASET

Feature/ Attribute	Description	Weight value
mol_wt	Molecular weight of the protein	1.880
theo_pI	Theoretical pI (ioselectric point)	0.325
atomic_comp_c	Atomic composition of C	9.348
atomic_comp_h	Atomic composition of H	5.558
atomic_comp_n	Atomic composition of N	9.544
atomic_comp_o	Atomic composition of O	8.059
atomic_comp_s	Atomic composition of S	8.714
aliphatic_index	The aliphatic index	0.456
hydro	Grand average of hydropathicity	8.469

strand	The DNA strand on which the ORF lies	9.371
position	Number of exons (how many start positions are there in its coordinates list).	0.232
motifs	Number of motifs: according to PROSITE dictionary release 13 of Nov. 1995	0.035
transmembrane spans	Number of transmembrane spans	0.104
chromosome	Chromosome number for this ORF	1.828

A descriptive illustration of attributes and the weights assigned by the genetic algorithm for the "seq" dataset are given in the TABLE II and we have evaluated 3 datasets for missing data imputation. All these attributes illustrated in TABLE II contains numerical values. Depending on the size of the data sets we have used missing rates ranging 10% - 30% in the three datasets. Fig. 5, Fig. 6 and Fig. 7 illustrate the mean errors calculated foreach imputation algorithm: meanImpute, kNNImpute and EvlkNNImpute. We have tested the missing value imputation of kNNImpute with the optimum *k* value obtained from EvlkNNImpute and with the *k* value of 10. TABLE III shows how the results vary with the different values for *k*.Even though the k value of kNNImputation is increased to 10, the mean error by the EvlkNNImputation got the lowest mean error by outperforming the kNNImputation.

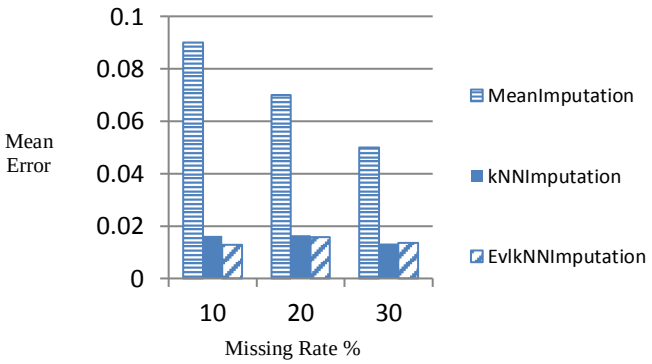

Fig.5. Mean errors of Mean Imputation, kNNImputation and EvlkNNImputation at different missing value rates. Algorithms imputedthe missing values of the attribute which indicates the 5 minutes heat shock effect in gasch2 dataset.

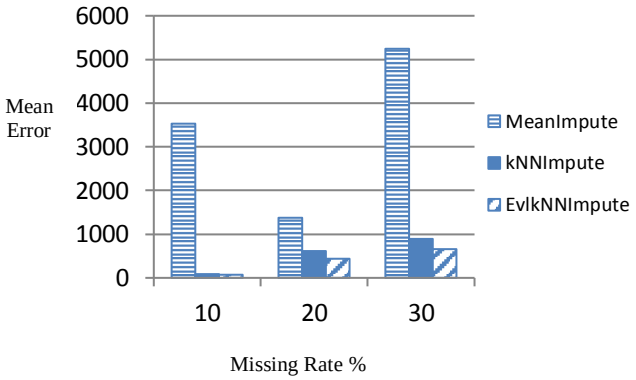

Fig.6. Mean errors of Mean Imputation, kNNImputation and EvlKNNImputation at different missing value rates. Algorithms imputed missing values in attributesof microarray expression data in spo dataset.

IV. Discussion and Conclusion

By looking at Fig. 5, Fig. 6 and Fig. 7we can conclude that the mean error from mean imputation is not effective and the error rates are relatively higher than the supervised algorithms. Also, by mean imputation the correlation of attributes in the data set is not taken into consideration by distorting the distribution and underestimating the standard deviation.

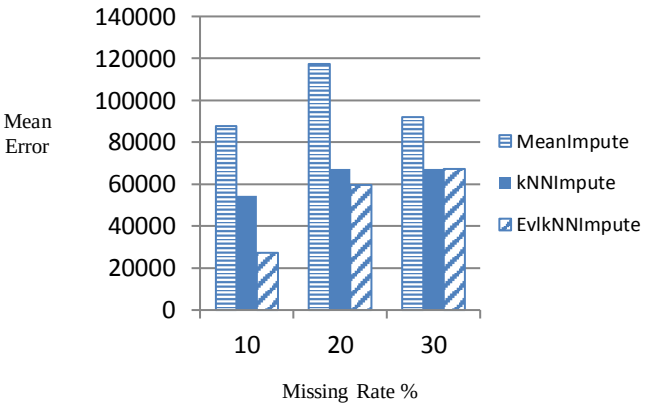

Fig.7. Mean errors of kNNImpute and EvlKNNImpute at different missing value rates. Algorithms imputed missing values in attribute which indicates molecular weight in seq dataset.

TABLE III
MEAN ERROR AT DIFFERENT MISSING RATES AND K VALUES

	gasch 2 dataset		
Missing Rate %	**kNN Imputation**	**kNN Imputation**	**EvlkNN Imputation**
	k=10	**k=5**	**k=5**
10	0.168	0.016	0.013
20	0.159	0.016	0.016
30	0.150	0.013	0.013
	spo dataset		

Missing Rate %	kNN Imputation	kNN Imputation	EvlkNN Imputation
	k=10	k=5	k=5
10	74.356	85.036	67.857
20	456.426	616.741	432.061
30	727.434	899.284	664.682

This is where the need of a supervised learning algorithm occurs and the proposed algorithm is designed to overcome the disadvantages in the kNNImpute algorithm. Though kNNImpute has been widely used for missing value imputation, there are many drawbacks when performing on a large dataset. EvlkNNImpute has the ability to identify the optimum value for *k* and assign weights to each attribute in the dataset.The figures also showthe mean errors of three imputation methods and EvlkNNImputation has the lowest mean error out of the three methods used in this paper. Also, by looking at the figures we can conclude that EvlkNNImpute performs better when there is a certain high level of missing rate in a dataset than a small missing rate in a dataset.

By observing the weights it can be clearly seen that the most important predictive attributes in predicting theclass labels are getting the higher weight values. The attributes such as the atomic composition of Carbon, Hydrogen, Nitrogen, Oxygen, Sulphur, and average of hydropathicity of the "seq" data set play a vital role when predicting the missing percentages of amino acids in certain proteins and the evolutionary algorithm has been successful in assigning higher weights for the above attributes as shown in TABLE II.

The attributes displayed in TABLE II can be categorized under the local correlation data and such data is effective towards the kNNImputaion and EvlkNNImpuation algorithms. The learning capability of the algorithm can be clearly seen by identifying the attributes mentioned above as important attributes to assign higher weights. Due to the drastic difference between the mean errors of mean imputation and the two supervised learning algorithms, we can conclude that the imputation of missing data in gene expression data sets need a supervised learning algorithm which will look for the correlation of attributes in genes. Furthermore, the mean error of EvlKNNImpute is lower than the kNNImpute because of the optimization methods developed using the genetic algorithm in order to overcome the disadvantages of the standard kNNImpute algorithm by identifying the optimum value for *k* and by assigning weights to attributes.

REFERENCES

[1] O. Troyanskaya *et al.*, "Missing value estimation methods for DNA microarrays," *Bioinforma. Oxf. Engl.*, vol. 17, no. 6, pp. 520–525, Jun. 2001.

[2] K. Moorthy, M. S. Mohamad, and S. Deris, "A Review on Missing Value Imputation Algorithms for Microarray Gene Expression Data," *ResearchGate*, vol. 9, no. 1, pp. 18–22, Jan. 2014.

[3] T. D. Pigott, "A Review of Methods for Missing Data," *Educ. Res. Eval.*, vol. 7, no. 4, pp. 353–383, Dec. 2001.

[4] S. Friedland, A. Niknejad, M. Kaveh, and H. Zare, "An Algorithm for Missing Value Estimation for DNA Microarray Data," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, 2006, vol. 2, pp. II–II.

[5] "Gene expression," *Wikipedia, the free encyclopedia*. 09-Jan-2016.

[6] J. L. DeRisi, V. R. Iyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, no. 5338, pp. 680–686, Oct. 1997.

[7] C. M. Perou *et al.*, "Molecular portraits of human breast tumours," *Nature*, vol. 406, no. 6797, pp. 747–752, Aug. 2000.

[8] S. Chu *et al.*, "The transcriptional program of sporulation in budding yeast," *Science*, vol. 282, no. 5389, pp. 699–705, Oct. 1998.

[9] A. Gelman and J. Hill, "Missing-data imputation," in *Data Analysis Using Regression and Multilevel%2FHierarchical Models*, Cambridge University Press, 2006.

[10] A. Mockus, "Missing Data in Software Engineering," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. Springer London, 2008, pp. 185–200.

[11] E. Acuña and C. Rodriguez, "The Treatment of Missing Values and its Effect on Classifier Accuracy," in *Classification, Clustering, and Data Mining Applications*, D. D. Banks, D. F. R. McMorris, D. P. Arabie, and P. D. W. Gaul, Eds. Springer Berlin Heidelberg, 2004, pp. 639–647.

[12] J. Kaiser, "Algorithm for Missing Values Imputation in Categorical Data with Use of Association Rules," *ArXiv12111799 Cs*, Nov. 2012.

[13] J. C. Wayman and P. D, *Multiple Imputation for Missing Data: What is It and How Can I Use It*. 2003.

[14] A. W.-C. Liew, N.-F. Law, and H. Yan, "Missing value imputation for gene expression data: computational techniques to recover missing data from available information," *Brief. Bioinform.*, vol. 12, no. 5, pp. 498–513, Sep. 2011.

[15] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii, "A Bayesian missing value estimation method for gene expression profile data," *Bioinforma. Oxf. Engl.*, vol. 19, no. 16, pp. 2088–2096, Nov. 2003.

[16] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, Jan. 2005.

[17] X. Zhang, X. Song, H. Wang, and H. Zhang, "Sequential Local Least Squares Imputation Estimating Missing Value of Microarray Data," *Comput Biol Med*, vol. 38, no. 10, pp. 1112–1120, Oct. 2008.

[18] R. Jörnsten, H.-Y. Wang, W. J. Welsh, and M. Ouyang, "DNA microarray data imputation and significance analysis of differential expression," *Bioinforma. Oxf. Engl.*, vol. 21, no. 22, pp. 4155–4161, Nov. 2005.

[19] Q. Xiang *et al.*, "Missing value imputation for microarray gene expression data using histone acetylation information," *BMC Bioinformatics*, vol. 9, p. 252, 2008.

[20] G. Batista and M. C. Monard, "A Study of K-Nearest Neighbour as an Imputation Method," in *In HIS*, 2003.

[21] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, 2007.

[22] P. Clark and T. Niblett, "The CN2 Induction Algorithm," *Mach Learn*, vol. 3, no. 4, pp. 261–283, Mar. 1989.

[23] C. Zhang, J. Kai, H. C. Feng, and T. Yang, "The Nearest Neighbor Algorithm of Filling Missing Data Based on Cluster Analysis," *Appl. Mech. Mater.*, vol. 347–350, pp. 2324–2328, Aug. 2013.

[24] K. O. Cheng, N. F. Law, and W. C. Siu, "Use of biclustering for missing value imputation in gene expression data," *Artif. Intell. Res.*, vol. 2, no. 2, Feb. 2013.

[25] P. Keerin, W. Kurutach, and T. Boongoen, "Cluster-based KNN missing value imputation for DNA microarray data," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2012, pp. 445–450.

[26] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector

regression and a genetic algorithm," *Inf. Sci.*, vol. 233, pp. 25–35, Jun. 2013.

[27] H.-H. Li, F.-F. Shao, and G.-Z. Li, "Semi-supervised imputation for microarray missing value estimation," in *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2014, pp. 297–300.

[28] D. J. Sobajic, *Neural Network Computing for the Electric Power Industry: Proceedings of the 1992 Inns Summer Workshop*. Psychology Press, 2013.

[29] C.-F. Tsai, W. Eberle, and C.-Y. Chu, "Genetic algorithms in feature and instance selection," *Knowl.-Based Syst.*, vol. 39, pp. 240–247, Feb. 2013.

[30] S. M. Thede, "An Introduction to Genetic Algorithms," *Journal of Computing Sciences in Colleges*, vol. 20 Issue 1, pp. 115–123, Oct. 2004.

[31] A. S. Perera and W. Perrizo, "Gene Function Prediction.," in *ResearchGate*, 2009, pp. 26–31.

[32] M. Middlemiss and G. Dick, "Design and Application of Hybrid Intelligent Systems," A. Abraham, M. Köppen, and K. Franke, Eds. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2003, pp. 519–527.

[33] S. M. Thede, "An Introduction to Genetic Algorithms," *J Comput Sci Coll*, vol. 20, no. 1, pp. 115–123, Oct. 2004.

[34] A. Clare and R. D. King, "Predicting gene function in Saccharomyces cerevisiae," *Bioinformatics*, vol. 19, no. suppl_2, p. ii42-ii49, Sep. 2003.