electronics

MDPI

Article

A Neighborhood-Similarity-Based Imputation Algorithm for Healthcare Data Sets: A Comparative Study

Colin Wilcox [1], **Vasileios Giagos** [2] **and Soufiene Djahel** [3,*]

[1] Department of Computing and Mathematics, Manchester Metropolitan University, Manchester M15 6BH, UK; colin.r.wilcox@stu.mmu.ac.uk

[2] Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, UK; v.giagos@essex.ac.uk

[3] Centre for Future Transport and Cities, Coventry University, Priory Street, Coventry CV1 5FB, UK

* Correspondence: ae3095@coventry.ac.uk

Abstract: The increasing computerisation of medical services has highlighted inconsistencies in the way in which patients' historic medical data were recorded. Differences in process and practice between medical services and facilities have led to many incomplete and inaccurate medical histories being recorded. To create a single point of truth going forward, it is necessary to correct these inconsistencies. A common way to do this has been to use imputation techniques to predict missing data values based on the known values in the data set. In this paper, we propose a neighborhood similarity measure-based imputation technique and analyze its achieved prediction accuracy in comparison with a number of traditional imputation methods using both an incomplete anonymized diabetes medical data set and a number of simulated data sets as the sources of our data. The aim is to determine whether any improvement could be made in the accuracy of predicting a diabetes diagnosis using the known outcomes of the diabetes patients' data set. The obtained results have proven the effectiveness of our proposed approach compared to other state-of-the-art single-pass imputation techniques.

Keywords: healthcare; imputation algorithms; incomplete data; neighborhood similarity

check for updates

Citation: Wilcox, C.; Giagos, V.; Djahel, S. A Neighborhood-Similarity-Based Imputation Algorithm for Healthcare Data Sets: A Comparative Study. *Electronics* **2023**, *12*, 4809. https://doi.org/10.3390/electronics12234809

Academic Editors: Wentao Li, Huiyan Zhang, Tao Zhan and Chao Zhang

Received: 6 October 2023
Revised: 22 November 2023
Accepted: 23 November 2023
Published: 28 November 2023

1. Introduction

Due to widespread computerization, medical services have embarked on moving their historic paper-based medical data onto computer systems [1]. This has raised a number of technical and societal issues. Generations of paper-based medical records need to be digitally encoded in a way that is not only capable of handling the large information backlog, but must also be accurate, sensitive, and, most importantly for many financially stretched services, cost effective [2,3]. Historic medical data has highlighted the inconsistencies of the previous recording and transcription practices and processes used by both medical practitioners and regional authorities such that, in many cases, data may be incomplete, incorrectly encoded, or just erroneous. This is not just a legacy issue, as modern recording techniques also suffer from similar issues of data incompleteness that emphasize the need to find a robust solution to this wider problem [4,5].

In the future, legacy data will form the basis of a much wider medical profile describing an individual and will include more granular and real-time information. This data may include a person's movements, access to medical facilities, data from personal fitness trackers, and other biometric devices. Data from all such sources need to be recorded in a consistent manner. By ensuring high quality and the accuracy of such data, these medical data sources become points of truth when identifying the individual to which they relate and can thereby be used as a means of individual identification.

Imputation is the overarching term used for describing the range of techniques used to replace missing data in a data set. The techniques can range from very simple numerical replacement to more complex statistical approaches. They can be broadly split into several types of approaches [6]:

- **Normal imputation**: When the data is numerical, we can use simple techniques, such as mean or modal values for a feature, to fill in the missing data. For data that is more categorical (i.e., they have a defined and limited range of possible values), then the most frequently occurring modal value for this feature can be used.
- **Class-based imputation**: Instead of replacing missing data with a calculated value based on existing feature values above, the replacement is done based on some internal classification. This approach determines the replacement value based on the values of a restricted subclass of known feature values.
- **Model-based imputation**: A hybrid approach where the missing value is considered as the class, and all the remaining features are used to train the model for predicting values.

The problem we aim to address concerns the rapidly growing amount of incomplete personal medical data that exists. The rapid increase in volume and complexity of this data has highlighted potential problems and issues caused by our current reliance on this incomplete or inaccurate information. Such unqualified use may lead to a loss or misinterpretation of critical medical information. This problem is not limited to a medical domain and equally applies to any problem domain that uses incomplete personal information in a technology-driven environment. The focus of this paper is on a medical context, but the solution should be readily generalizable to other problem domains. The existence and use of incomplete medical data may lead to a loss or misrepresentation of critical medical information [6]. The increasing amount and variety of stored data about individuals in the smart healthcare era only emphasizes the urgency in finding solutions to this problem [7]. Our approach will select imputed data values in a more localized manner, thus applying a more intelligent selection of candidate values rather than one of the more simplistic, and widely used, imputation methods.

In this paper, we propose a neighborhood-based imputation algorithm that uses the idea of feature value similarity in similar data records to predict missing feature values in incomplete records. This subset of candidate records is specific to a single incomplete record and so is recreated for each incomplete record found in a data set. This differs from other imputation techniques, which may consider all records in the data set and give a more general and less localized result, or other approaches, which determine neighborhood values based on other criteria such as using weighted average or variance estimation techniques [7].

Our algorithm aims to improve on some of the limitations of existing imputation algorithms, especially kNNs, by providing a fast, yet accurate imputation process suitable for use on, initially, medical data, but also on more generic incomplete data sets from other similar problem domains. The main contributions of this work can be summarized as follows:

- Reducing the speed degradation of the algorithm as the size of the data set increases.
- The way imputed values are selected is more localized rather than potentially using all similar values in the data set.
- Reducing the negative impact of outlying values by making imputed values selection more localized.
- Providing a solution that can be extended for use with textual and categorical data, as well as numeric data.

The remainder of this paper is organized as follows. In Section 2, we present the background to understanding the problem being studied in this paper. Section 3 presents our proposed algorithm to improve prediction accuracy, and Section 4 evaluates the performance of our proposed technique in comparison with other imputation methods. Section 5 discusses our conclusions and findings during this work, and, finally, Section 6 indicates some directions for future work.

2. Background and Related Work

In this section, we present the background of incomplete medical data and the reasons why data integrity and completeness are important.

Imputation is the name given to the range of techniques that attempt to restore missing information in a data set with values based on the feature values of complete data records. The complexity of this process can range from merely replacing missing values with fixed absolute values, thus applying some mathematical function to known feature values for a given missing feature, or, in the simplest scenario, incomplete data records may be completely removed from consideration [6,8]. The choice of the technique used depends on a number of factors, including the nature of the source data, the amount of missing or erroneous data in the data set, and the time needed to create a suitably complete set of data. More complex approaches, such as time-series-based methods, attempt to rebuild potential structures within the data set by considering wider factors such as patterns in the data and relationships between the values of related features rather than just individual value replacement. Examples of such approaches include linear interpolation techniques, which take two known feature values and use a weighted distance between these endpoints to calculate intermediate values [9], and the use of adjacent known feature values as candidates for replacing missing feature values [10]. Such techniques tend to be more time consuming, and their effectiveness is reliant upon the intended use and ability to identify suitable structures to recreate within the source data [11]. Many of these restoration techniques have analogies in the non-digital world, which may be considered as possible approaches for imputing sets of data. In the following section, we briefly discuss three common approaches to single-pass imputation [12].

2.1. Imputation by Mean/Mode/Median and Others

If the missing values in a data set's feature column are numeric, they can be imputed by using the mean value of the existing values for that feature variable. The mean imputed value could be replaced by the median feature value if the feature is suspected to have outlying values. For a categorical feature, the missing values could be replaced by the mode of the existing values for that feature. The major drawback of this method is that it reduces the variance of the imputed variables. This method also reduces the correlation between the imputed variables and other variables, because the imputed values are just estimates and will not be related to other values inherently [13].

Another algorithm worthy of note is the k-nearest neighbors (kNNs) algorithm [14]. In a similar manner to our proposed algorithm, kNNs attempts to impute missing feature values by using the mean value of the corresponding known feature values for the k-closest records. The kNNs algorithm has a number of limitations, which our algorithm attempts to resolve. The kNNs is a robust algorithm belonging to a family of *nearest neighbor* algorithms used to predict unknown classifications based on a data set of known classifications. It is commonly used because it is intuitive and easy to implement and is nonparametric, meaning that it makes no prior assumptions about the nature of the data set. It may be used for both classification and regression problems, thus making it a widely used and popular choice of algorithm.

The kNNs algorithm has a number of disadvantages, which our solution attempts to improve upon and include the following:

- The kNNs is a relatively slow algorithm, with its performance decreasing as the size of the data set increases.
- The kNNs suffers from the curse of dimensionality [15]. As the number of feature values (dimensions) per record increases, the amount of data required to predict a new data point increases exponentially.
- The manner in which kNNs measures the closeness of a pair of records is quite simple, by using Euclidean or Manhattan distances for example.
- The kNNs algorithm needs homogeneity such that all the features must be measured using the same scale, since the distance is taken as an absolute measure.

- The kNNs does not work well with imbalanced data. Given two potential choices of classification, the algorithm will naturally tend to be biased towards a result taken from the largest data subsets, thus leading to potentially more misclassifications.
- The kNNs is sensitive to outlying values, as the choice of closest neighbors is based on an absolute measure of the distance.

Our algorithm aims to improve on these drawbacks, especially in the areas of outlier sensitivity, thereby reducing the likelihood of misclassification and the choice of imputed feature values. Since the kNNs uses the mean of the k-nearest feature values, this could lead to a value being calculated that does not appear in any of the actual complete records; our algorithm removes this scenario by only choosing imputed feature values from a pool of candidate values taken from the actual feature values of the most similar complete records.

The class of nearest neighbour predictive algorithms can make accurate predictions, which do not require a human-readable model [16]. The quality of these predictions depends on the measure of the distance between the data values [17]. There are several advantages to this class of algorithms, including a robustness for noisy data and the ability to be tuned quite easily. However, the kNNs has some drawbacks, such as the need for all the feature values for any missing value to be considered. This was a motivation and opportunity to use a more localized approach for determining missing data values [16].

2.2. Simple Statistical Imputation Techniques

Statistical techniques are usually applied because they tend to be fast, have low memory overhead, and are applicable in isolation to any surrounding data. These simpler approaches involve determining the value of a missing feature by applying a simple functional calculation on the set of known feature values [15]. In our comparison, these are represented by the mean (MAV) and modal (MDAV) value algorithms. Calculations tend to be linear in nature and applied independently from other data fields in the same data set. Calculations may range from setting missing data values to a known fixed value to finding an average of those values that exist in other records in the data sets, or some trivial manipulation of existing data values from other records [18]. More involved algorithms have been developed, which try to use wider information about the nature of the data values and any relationships that may exist between features as a way of more accurately determining missing feature values. In our discussion, we highlight two such algorithms, kNNs imputation and empirical Bayes inference; however, there are many more that could be considered. This approach can be extended to use multiple imputation techniques, which involves repeatedly applying simple mathematical techniques to improve the missing feature value prediction, as defined by the pseudo flow below:

1. Identify missing values in the source data set.
2. Iterate through the data set. For each record with missing values, replace each missing value with a statistical measure based on values for the same field found in other records where this field is not missing.
3. Once all the records have been completed, if the nature of the data set meets the criteria for its intended use, then stop; otherwise, repeat Step 2.

2.3. Multistage Techniques

Multiple imputation is a general approach to the problem of missing data that is available in several commonly used statistical packages such as R [19,20]. Single-pass imputation is the process of "filling in" gaps representing missing values in data sets. An imputation method is a function that takes a number of known feature values as inputs and uses them to calculate a potential value for a missing feature value. Single-pass imputations apply such a mapping only once to the original set of known feature values. Multiple imputation, however, is a technique for reducing the uncertainty of missing values in a data set by creating several different viable imputed data sets and appropriately combining the results obtained from each of them to determine a suitable replacement value. We will compare the performance of our N-Similarity (NSIM) algorithm against that of three simple single-pass imputation algorithms, which either replace the missing feature value with the mean (MAV) and modal (MDAV) values of the known feature values or just remove all incomplete records from the processed data set.

Using single values carries with it a level of uncertainty about which values to impute. Multiple imputation reduces this uncertainty by calculating several different possible values ("imputations"). Several versions of the incomplete data sets are created, which are then combined to make the "best" value selections. Such an approach has several advantages such as reducing bias and minimizing the likelihood of errors being introduced to the rebuilt data sets, thus improving the validity of the data and increasing the precision or closeness between two or more imputed values, which makes the data set more resistant to outlying values [21,22].

The second stage is to use common statistical methods to fit the model of interest to each of the imputed data sets. Estimated associations in each of the imputed data set will differ because of the variation introduced in the imputation of the missing values, and they are only useful when averaged together to give overall estimated associations. Valid inferences are obtained because we are averaging over the distribution of the missing data given the observed data [23,24].

Other data-focused approaches using machine learning and deep data analysis techniques are being used as a means of predicting medical events from incomplete medical data sets. The use of such automated tools in the identification and prediction of medical conditions is becoming increasingly important due to the shortage of skilled medical professionals, as well as their ability to increase the prediction accuracy, thus reducing the burden on medical staff [25,26].

3. Proposed Algorithm

In this section, we outline our approach to improving the effectiveness of predicting binary outcomes based on a series of numerical feature values. We used a suitably anonymized diabetes diagnosis data set, which identified whether a patient with diabetes has been positively diagnosed (*true positive*) or whether one who does not have diabetes has been negatively diagnosed (*false positive*).

3.1. Proposal Main Steps

Our algorithm aims to improve on a number of traditional single-pass imputation techniques to achieve a higher percentage of correct predictions when applied to an incomplete diabetes data set, D. The approach will consist of the following steps.

- Apply our imputation technique to fill in each missing attribute f_i in turn, where i corresponds to the ith feature in each patient record, for the current record r to create a complete record in D. This will become the basis of the later comparisons. Incomplete records r are given by

$$\forall r \in D, r = (f_0, f_1, f_2, \ldots f_i - 1, f_i + 1, \ldots) \quad (1)$$

- Use the k-fold (with k = 10) [27,28] technique to partition D into non-intersecting subsets. In turn, each subset (fold) will be considered to be the *test fold*, and the remaining folds will be used as *training folds*. For each record in the test fold, we apply a comparison function $F()$, which is in our case the cosine similarity, to obtain a numerical measure of how similar the test record is to the current record in the training folds. An ordered *similarity table*, S, is maintained and stores details of each training record and how similar it is to the current test record. This is repeated until the test record has been compared against all the records in all the training folds. After each change to the contents of S, it will be sorted in such a way that the most similar training record will appear as the first item in the list. This could be more complicated depending on the comparison function used, but in our case, the sort order is merely used to maintain the n-closest items (defining the neighborhood) in S in an increasing cosine similarity order. The contents of S must be cleared once all the training set records have been compared and are ready for subsequent cycles.

Folds containing a large number of records can increase the time needed to compare all the combinations of these records against a given test record. This could result in a relatively large similarity table. To address this issue of similarity table size, our proposed algorithm introduces the concept of a *neighborhood* containing the most similar n records in the training set. The size of this neighborhood limits the maximum size of the similarity table and is used as a means of calculating the new replacement value for a missing attribute.

Considering S_t to be the set of test records and S_{tr} to be the set of training records for a given cycle, such that $t \in S_t$ and $tr \in S_{tr}$, we can say that

$$\forall t \in S_t, \forall tr \in S_{tr}; S_t \cap S_{tr} = 0, S_t \cup S_{tr} = D \tag{2}$$

If there are less than n records in the similarity table, then add the current training record, tr, into the next freely available position p. If the similarity table already contains n records and the current test record, t is more similar than the last record in the similarity table (at position $n - 1$ for zero-based arrays); then, we replace the last entry in the similarity table with the current training record tr. This can be shown with the pseudocode below.

```
clear SimilarityTable, S
FOR EACH t IN testFold DO
    p <- 0
    FOR EACH tr IN trainingFolds DO
        size = count(S)
        IF size < n THEN
            S[p] <- F (t, tr)
        ELSE
            IF F(t, tr) > S[n-1] THEN
                S[n-1] <- F(t,tr)
```

Each time the contents of the similarity table are changed, they should be immediately sorted based on decreasing similarity value to maintain a list of the most similar training records for the current test record. In order to build a complete data set D, we need to calculate each of the missing data values across all the records in D. This is achieved by comparing each row that contains missing values against all the complete rows that exist in D. By doing this, we build up a similarity table containing the most similar complete records from which the candidate values for the missing data values may be selected. Once all the complete records in the data set have been compared against the current incomplete record, we are in a position to impute the missing values for the current record in order to make it complete. This record can then be used as a candidate record for matching the other incomplete records in later cycles of the process. The end result will be a completely imputed data set, which can then be used for comparison purposes with the different imputation techniques.

3.2. Similarity Model Behavior

Our proposed model is built around the idea that patients with the same sets of symptoms (features) will result in the same diagnosis. A patient with an unknown diagnosis will have a number of recorded symptoms, which may or may not be complete. Our algorithm takes those features that are known and uses them to find those diagnosed patients (neighborhood) that are the closest match in terms of the most similar features. This neighborhood is then used to determine what the likely diagnosis of the target patient may be. This has the advantage over other techniques in the fact that only similar patient records are used to build the picture of the diagnosis rather than a much wider spread of patients who may have less correlation with the patient in question.

This similarity model is based on the splitting of the source data set as previously described. The idea is to take the source data set and split it into two disjoint subsets—the training data set and the test data set. The splitting of the source data ensures that the number of records in the test data set is a fixed proportion of the total number of records according to the supplied parameters.

Each record in the test data subset is compared in turn with each record in the training data subset. A comparison of each pair of records is made using the concept of cosine similarity to obtain a measure of how similar the corresponding pairs of field attributes are with each other, thus yielding a numeric measure of their similarity. During this process, a similarity table is built giving a similarity measure of each training record in the training set against a single test record. This table is maintained such that the record with the most similar value (i.e., the most similar) is the first record in the table. The rationalization is that the training set records that are considered to be a similar match to the test record, and therefore the initial best-matching training record, will have very similar values for their input arguments, and, as such, they are the best candidates to determine whether the outcome given by the closest-matching test record was in fact valid.

Finally, a replacement value for the missing attribute, f_i, is determined by applying a prioritized set of rules to choose the most appropriate value from the candidate value set C. This approach may be extended to include '*categorical variables*', which describe those features that take a value from a limited set of possible values. Since the feature value set C, used as the pool of possible replacement values, is constructed from known feature values of the most similar records, then the selection rules are equally applicable and will select a suitable replacement value from C.

Considering the process diagram shown in Figure 1, the similarity modeling process is split into two main subflows. The colors used are unimportant and just used for highlighting purposes. The blue flow describes the processing steps of loading external data and standardizing it into a form that can be used by the second (green) flow through the application of the k-fold technique to split the source data set into folds. The green flow indicates the application of the N-Similarity algorithm. The key points of the algorithm flow are to take each fold as a test record in turn and apply cross correlation against each of the remaining training folds to generate the similarity table of the most similar training records for each record in the test fold. This is repeated for each training record until all comparison combinations have been performed. For each incomplete record, the missing feature value is determined by considering the properties of the closest records in the similarity table, and a candidate is selected based on a number of rules and criteria. The results of these comparisons are shown in Table 1.

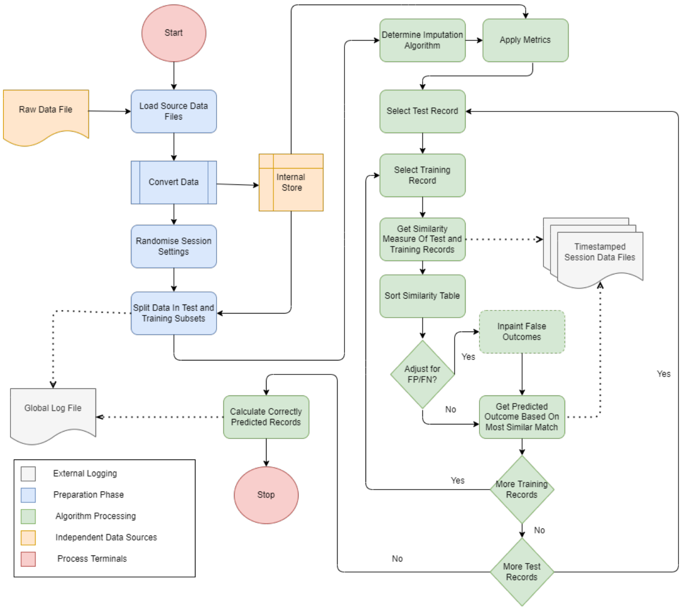

Figure 1. Main steps of the similarity modeling process.

Table 1. Relative prediction accuracy of our N-Similarity algorithm compared to the average prediction accuracy across all selected single imputation techniques for different neighbourhood sizes N.

	N = 1	**N = 2**	**N = 3**	**N = 4**	**N = 5**	**N = 6**	**N = 7**	**N = 8**	**N = 9**	**N = 10**
Accuracy	55.64%	73.37%	58.01%	69.84%	58.84%	70.82%	60.13%	66.81%	60.74%	67.41%
Correlation	76.97%	88.91%	89.65%	89.55%	89.57%	89.06%	89.30%	89.63%	89.33%	88.99%
Precision	31.12%	58.41%	30.79%	55.59%	32.28%	59.07%	32.30%	46.98%	33.53%	48.58%
Recall	33.01%	55.13%	28.71%	39.62%	25.50%	29.73%	23.53%	32.57%	24.23%	28.60%
Specificity	66.18%	81.47%	71.34%	83.76%	74.22%	89.57%	77.04%	82.43%	77.49%	85.12%
TPR	23.03%	38.12%	20.34%	28.12%	18.19%	20.44%	16.48%	23.07%	17.10%	19.80%
FPR	33.82%	18.53%	28.66%	16.24%	25.78%	10.43%	22.96%	17.57%	22.51%	14.88%
Average MCC	**−0.0495**	**0.4582**	**0.0792**	**0.3419**	**0.0842**	**0.3069**	**0.0383**	**0.2326**	**0.0123**	**0.1771**

The colour coding scheme used in Table 1 reflects how, for different neighbourhood sizes, the prediction accuracy of our N-Similarity algorithm compares to the average prediction accuracy of the other imputation algorithms under consideration. The green values indicate those measures where our algorithm performs better than the average of the other imputation algorithms, red values indicate those measures where our algorithm performs worse, and the blue values indicate those measures where there is marginal difference between the algorithms.

3.3. Empirical Bayes Correction

Dealing with missing data and its mechanism is of paramount importance in statistics [29], and in this section, we propose a correction for imputing numerical variables motivated by a normal-normal hierarchical model (see [30], Section 3.3.1). Let $D = \{Y, X\}$ be our observations, where Y is the part that contains missing values, and X (a $N_{\text{Obs}} \times N_X$ matrix) is fully observed. We consider the following correction term for the imputed candidate value $\hat{\theta}_m$ given the (observable) sample mean $\bar{Y}$ and the most similar value Y^*_m:

$$\hat{\theta}_m = \alpha\bar{Y} + (1-\alpha)Y^*_m, \quad \alpha = \frac{s_y^2}{s_y^2 + (\hat{\tau}^2_{Y|X})^+}, \tag{3}$$

where s_y^2 is the sample variance of $y = (y_1, \ldots, y_l)$ for the l-most-similar observations (comparing X_m to X_{obs}), and $(\hat{\tau}^2_{Y|X})^+$ is an approximation of the Empirical Bayes estimate of [30]:

$$(\hat{\tau}^2_{Y|X})^+ = \max\left[0, \lambda \times s_Y^2 - s_y^2\right], \tag{4}$$

where λ is a fixed hyperparameter, and s_Y^2 the sample variance of the observable Y.

Since $0 \le \alpha \le 1$, and (3) is a weighted average between Y^*_m and $\bar{Y}$, which essentially shrinks the proposal towards the mean $\bar{Y}$, the amount of shrinking is determined by α. When $\alpha = 0$, (3) suggests a direct imputation with Y^*_m, whereas $\alpha = 1$ suggests an imputation using $\bar{Y}$. Generally, our candidate imputed value shrinks towards $\bar{Y}$ when the variance associated with Y^*_m exceeds the sample variance of Y.

Motivation

We motivate (3) by considering an empirical Bayes approach to our hierarchical model. We introduce two types of random variables: one expressing the missing values Y_m and one $\theta_{m|X}$ expressing the neighboorhood-similarity-based guesses (can also be thought of as model-based guesses) that rely on a relation between Y and X. For each missing value Y_m, we assume that it is a normal random variable with mean $\theta_{m|X}$ and a variance $\sigma^2_{m|X}$. This allows us to express the "true" missing value in relation to our similarity-based guesses: for ms with small variances ($\sigma^2_{m|X}$), the similarity-based guesses are informative, and for large variances, they are not.

For each $\theta_{m|x}$, we again assume a normal distribution with a common mean and variance $(\mu_{Y|X}, \tau^2_{Y|X})$:

$$Y_m \Big| X, \theta_{m|X}, \sigma^2_{m|X} \sim N\Big(\theta_{m|X}, \sigma^2_{m|X}\Big) \tag{5}$$

$$\theta_{m|X} \Big| X, \mu_{Y|X}, \tau^2_{Y|X} \sim N\Big(\mu_{Y|X}, \tau^2_{Y|X}\Big), \tag{6}$$

which expresses the overall relation of Y given X as a normal distribution with its mean and variance varying according to X. In other words, instead of considering the similarity-based guess of the missing value as a single point, we introduce a normal-distributed kernel centered around it, which depends on the fully observed X. Our two-level hierarchical model uses (5) locally to express the distribution of Y_m and (6) to express the associated mean $\theta_{m|X}$ using a global model between X and Y. Given a candidate value Y^*_m, we can impute Y_m with the posterior empirical Bayes mean $\hat{\theta}_{m|X}$ [30], which is a point estimate of $\theta_{m|X}$:

$$\hat{\theta}_m = \alpha\mu_{Y|X} + (1-\alpha)Y^*_m,$$

where $\alpha = \sigma^2_{m|X}/(\sigma^2_{m|X} + \tau^2_{Y|X})$. Linear and nonlinear regression models have been used for the conditional mean $\mu_{Y|X}$ in a Bayesian setting [31], whereas [32] used a nonparametric kernel regression, but in our performance evaluations, we also considered the weighted sample mean and sample variance, e.g., $s^2_y = \sum_i w_i(y_i - \bar{y})^2$, with weights approximated by a Gaussian kernel with a minimal RMSE improvement. The empirical Bayes estimate of [30] for $\tau^2_{Y|X}$ is based on sample estimates for $\sigma^2_{m|X}$ and $\tau^2_{Y|X}$:

$$(\tau^2_{Y|X})^+ = \max(0, \lambda\hat{\tau}^2_{Y|X} - \hat{\sigma}^2_{Y|X}).$$

If we consider the case that Y and X are independent, any similarity between X_{obs} and X_m provides no information about the missing Y_m. This also implies that $\mu_{Y|X}$ and $\sigma^2_{Y|X}$ become the marginal μ_Y and σ^2_Y, respectively. Furthermore, the y sample becomes a random sample of Y, with $\bar{y}$ and s^2_y being unbiased estimates of μ^2_Y and σ^2_Y, respectively. Therefore, we can use $\bar{Y}, s^2_Y$, and s^2_y as approximations for $\mu_{Y|X}, \hat{\tau}^2_{Y|X}$, and $\hat{\sigma}^2_{Y|X}$, respectively, which, under independence, set α towards one and can serve as a warning for noninformative imputation. Finally, if Y and X are not independent, y will be a conditional sample from $Y|X_m$, and we expect $\text{var}(Y) \geq \mathbb{E}[\text{var}(y)]$ to lead to to smaller shrinkage ($\alpha < 1$) towards $\bar{Y}$.

4. Performance Evaluation

In this section, we evaluate the performance of our similarity-based approach, using the sample diabetes data set, in comparison with a number of other imputation techniques.

4.1. Implementation Overview

The algorithm is made up of three steps: the *first step* is to partition the raw data set, D, into two disjoint subsets: one containing all the complete records, S_c, and the other containing records that are missing one or more feature values, S_i. The incomplete records are then checked in order. Whenever an incomplete record $S_i(k)$ contains a missing feature value $f_{k,i}$, the nearest N-Similarity algorithm (*second step*) is applied to create a similarity table of the closest n records from S_c. The missing feature value, $f_{k,i}$, is then determined by applying a series of rules below during the *third* and final step.

$$S_c \cup S_i = D, S_c \cap S_i = \varnothing$$

Considering the corresponding feature values of the n-most-similar complete records in the similarity table created by the stage above, the algorithm creates a set of candidate values, C, that will be used to replace the current missing feature value. The algorithm uses a number of simple rules, applied in strict order, to determine which of these candidate values is the *most likely* to be used as the replacement value for the missing feature in the current incomplete record.

$$\forall k \in S_i, f_{k,i} = S_c(j), 0 \leq jn$$

where j is the index of the best candidate value in C.

The set of rules applied to C in determining a predicted value are derived from both an evaluation of the corresponding feature values in the most similar diabetes records together with the nature of the values in the candidate set C. The rules are applied in order, with the most specific selection criteria applied first and moving down to the most general selection criteria applied last. For the candidate value set, C, apply the following rules in order of decreasing priority:

1. If there is a unique modal value in C, then use this value as the imputed feature value.
2. For those modal values which occur in C with equal highest frequency, if one of these modal values has the same feature value as the actual feature value of the most similar complete record in S_c, then select this modal value as the new imputed feature value for the current incomplete record.
3. Determine whether one of the values in C lies closer to the median value of the candidate set than the others. If such a value is found, select this as the imputed feature value.
4. If none of the previous rules have been satisfied, then select the mean value of C.

By comparing the prediction accuracy of the algorithm on the training data set (training folds), we can determine that the results are not noticeably different than the results obtained by applying the algorithm on the test data set (test fold), and therefore, we can ascertain that the algorithm does not overfit the diabetes data set.

This is repeated for each missing feature in the current partial record $S_i(k)$, after which the now complete record is moved from S_i to S_c to become a potential candidate for the completion of the next incomplete record in S_i.

4.2. Evaluation of RMSE

The evaluation was performed using a simulation-based approach that consists of repeatedly using a random selection of M records from the complete data records subset S_c. Since they were complete, each of these records had a known *actual* value for each feature, which could be used later for comparison purposes. The selected M values of each feature, f_i, were ignored and imputed using our N-Similarity algorithm in order to provide a more reliable estimate for the RMSE. These *predicted* values were then compared against the *actual* values to provide an estimate for the root mean squared error measure (RMSE) to determine the predictive performance of our algorithm [33]. In Sections 4.3 and 4.4.3, we used the three methods, i.e., similarity (NSIM), similarity with empirical Bayes correction (NSIM-EB), and k-nearest neighbors (kNNs) to repeatedly impute each feature and report the corresponding RMSEs.

4.3. Simulated Dataset

We proceeded to simulate 1000 datasets (of 1000 observations each) based on $x_1, \ldots, x_6$ (7) random variables. The x_1, x_2, and x_3 are independent Poisson, uniform, and exponential-distributed random variables, respectively, whereas the z_1, z_2, and z_3 are independent standard normal variables. The remaining (x_4, x_5, and x_6) are functions of the previous ones, with their relations outlined in (7). Overall, the simulated data sets contain an independent random variable (x_1), as well as noisy nonlinear relationships (e.g., x_6 with x_2).

$$\begin{aligned}
z_1, z_2, z_3 &\sim \text{Normal}(0,1) \\
x_1 &\sim \text{Poisson}(1) \\
x_2 &\sim \text{Uniform}(18,83) \\
x_3 &\sim \text{Exponential}(1/30) \\
x_4 &= z_1 \times x_3 + 3 \\
x_5 &= x_4 \times 3 + z_2 * \sqrt{10} \\
x_6 &= \exp(-x_2 \times 0.2 + z_3)
\end{aligned} \tag{7}$$

Table 2 shows the imputation RMSE of the three methods assuming 1.50 and 100 missing observations (M) per each simulated data set. Overall, the RMSE for the NSIM-EB was consistently lower than the rest. For x_1, as the number of M increased, the RMSE increased too for all the methods, which is expected, as x_1 is independent from the rest. Generally, the RMSE of the NSIM was similar, if not slightly reduced, compared to the RMSE of the kNNs. Both similarity-based methods were faster (NSIM performed in 126 s and NSIM-EB performed in 167 s; both were implemented in R) compared to the kNNs (215 s) using the implementation (with Mahalanobis distance) of the yaImpute package [34].

Table 2. Imputation of RMSE for simulated data using similarity (NSIM), similarity with empirical Bayes correction (NSIM-EB), and k-nearest neighbors (kNNs) methods.

M	Method	x_1	x_2	x_3	x_4	x_5	x_6
1	NSIM	1.382	1.402	1.414	1.378	1.345	1.333
	NSIM-EB	0.996	1.054	1.068	1.052	1.052	1.035
	kNNs	1.399	1.422	1.508	1.453	1.456	1.386
50	NSIM	1.421	1.401	1.398	1.402	1.401	1.380
	NSIM-EB	1.047	1.035	1.041	1.042	1.042	1.027
	kNNs	1.417	1.413	1.407	1.409	1.405	1.399
100	NSIM	1.420	1.396	1.385	1.386	1.385	1.373
	NSIM-EB	1.046	1.031	1.034	1.038	1.038	1.014
	kNNs	1.413	1.418	1.411	1.415	1.410	1.417

4.4. Pima Indians Diabetes Data Set

Another data set that was used extensively in this paper is the *Pima Indians Diabetes* data set [35], which is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The complete data set contains information of 768 women from a population located around Phoenix, Arizona, USA. The outcome tested was for diabetes, with 258 testing positive and 510 testing negative. The data was structured as follows: there was one target (dependent) variable and eight (feature) attributes: the number of pregnancies, oral glucose tolerance test, blood pressure, skin thickness, insulin, body mass index, age, and pedigree diabetes function. More technical details of the file used can be seen in Table 3. The Pima population has been under study by the National Institute of Diabetes and Digestive and Kidney Diseases at intervals of 2 years since 1965. As epidemiological evidence indicates that type 2 diabetes results from the interaction of genetic and environmental factors, the Pima Indians Diabetes data set includes information about attributes that could be related to the onset of diabetes and associated future complications.

The original data used zero as the marker for a missing feature value, because it was deemed that this could never be a valid value based on the nature of the features being represented. The obvious exception to this is the final binary outcome field, which may have a value of zero (for a negative diagnosis). The diagnosis outcome was a binary integer value indicated by a one for a positive diagnosis and a zero for a negative diagnosis, although in actuality, any nonzero integer would equally be interpreted as a positive diagnosis. Where it was possible, we converted this encoding convention to the programming language's standard missing value mechanism (e.g., Section 4.4.3), or we adapted our implementation (e.g., in Section 4.4.2, the similarity calculations are based only on valid feature values).

Out of the total number of records, 336 were complete (no missing feature values) (43.75%), and there were 763 missing feature values spread across the data set out of the total number of 6144 feature values (12.42%).

Table 3. Structure of PIMA diabetes data file.

Feature	**Data Type**	**Value Range (Zero Indicates Missing Value)**
Number of Times Pregnant	Positive Integer	0. . . 17
Plasma Glucose Concentration	Real	0. . . 199
Diastolic Blood Pressure	Real	0. . . 122
Triceps Skinfold Thickness	Real	0. . . 99
Serum Insulin Levels	Real	0. . . 846
Body Mass Index	Real	0. . . 67.1
Diabetes Pedigree Function	Real	0.078. . . 2.42
Age	Positive Integer	21. . . 81
Classification	Binary	1 = positive diagnosis, 0 = negative diagnosis

4.4.1. Comparison with Popular Imputation Methods

Three popular imputation techniques were used to provide a comparative baseline for the results obtained from applying our N-Similarity algorithm [36]. *Listwise deletion* is the process of removing all incomplete records from a data set prior to imputation [37]. If the original data is incomplete, then its application will naturally result in a smaller data set being produced for analysis. Depending on the sparsity of the original data, this may impact any ongoing analysis, thereby making it an unviable option for comparison against other imputation techniques that attempt to restore missing feature values without removing data. The statistical power [38] relies in part on a high sample size, and this is helped by having a relatively complete data set with few incomplete records. The other possible drawback to using listwise deletion is when the missing feature values may not be randomly distributed. For example, this occurs if a certain feature has missing values based on the nature in which the values for that feature were collected (questions aiming to extract sensitive information that the individual just skipped). As a result, and again depending on the level of sparsity of such missing data, the results may introduce bias into later analysis. One possible way to address these limitations and reduce the bias is to use multiple imputation techniques [39,40]. An extension of this approach, which was considered as a technique for comparison, was *pairwise deletion* [41]. This approach allows for the use of incomplete data but only allows for analysis on those features that have complete data. This introduces bias and makes like-for-like analysis more difficult, so it was was rejected as an option.

Some analysis has been undertaken [42] to determine the most popular imputation methods since 2000 (Figure 2). *Popularity* has been measured based on the number of times each imputation algorithm is mentioned in Google Scholar articles and papers. The results are somewhat surprising, since simpler, older techniques seem to be more popular than more recent approaches:

- **Remove Incomplete Records (Listwise Deletion)**: Any records in D that have one or more missing feature values are removed from the data set prior to processing. The removal of any incomplete records will lead to a smaller but complete data set D. It is not recommended that this technique is used arbitrarily as a means of direct comparison with other techniques used in the paper, since factors, such as the initial completeness of D, need to be assessed. It has been included due to its general popularity only (Figure 2).

- **Replace Missing Data With Mean Attribute Value**: Any missing feature values are replaced with the *average* value calculated from the corresponding feature values in all the complete records in the data set.
- **Replace Missing Data With Modal Attribute Value**: Any missing feature values are replaced with the *most common* value gathered from the corresponding feature values from all the complete records in the data set.
- **Replace Missing Data Using Empirical Bayes Algorithm**: This method is for statistically inferring missing feature values using a prior distribution of known values in a data set.
- **Replace Missing Data With N-Similarity Algorithm**: Any missing feature values are replaced with the *best candidate* value calculated from the corresponding feature values in the *N*-most-similar complete records in the data set.

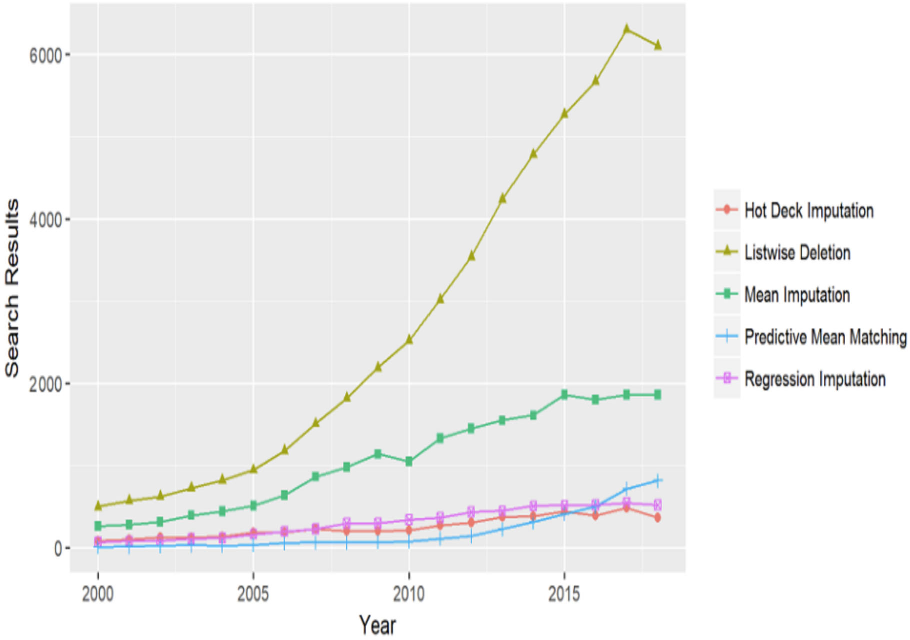

Figure 2. Google Scholar search results (Statistics Globe 2019).

Cross validation is a sampling procedure used to evaluate models that use a limited data sample. The procedure has a single parameter called *k* that refers to the number of equal sized groups (or folds) over which the data sample will be equally divided. The procedure is often called *k-fold cross validation* and is used to estimate the ability of a machine learning model to make predictions based on unseen data; it uses a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

The average cross validation over *n* folds is given by

$$\frac{1}{n}\sum_{k=1}^{k=n} Similarity_k$$

where $Similarity_k$ is the measure of similarity between the current test and the training folds for the session run *k*.

4.4.2. Results, Limitations, and Discussion

Table 1 shows the improvement in predicting true positive cases when using our proposed N-Similarity algorithm compared against several other single-pass imputation algorithms. Entries highlighted in *green* show the improvement achieved by our algorithm over other popular techniques. Those highlighted in *red* show a worsened prediction accuracy. Those entries highlighted in *blue* indicate no or negligible change in the prediction accuracy.

The testing process splits the data sets into **ten** approximately equally sized folds. The arbitrary partitioning of records in each fold, in any given run, meant that each fold could contain a combination of complete and incomplete records. The proportion of incomplete records was allowed to vary so as to not impose any potentially restrictive classification on the fold contents. Should we have wanted to impose a limiting proportion of incomplete records in a fold, for some reason, then a stratified k-fold approach or similar would have been used. When applying cross correlation techniques, some of the ratio calculations shown in Table 1 had no correctly predicted positive outcomes ($TP = 0$), thus leading to incomplete runs being produced. Similarly, in some folds, it may also be possible that the number of true positive (TP) and true negative (TN) training records are not predictable for a given fold, thus meaning that the *precision* metric was indeterminate for specific pairings of test and training data folds, since $TP + TN = 0$. The likelihood of these eventualities could be reduced somewhat by reducing the number of folds for the given data sets, thereby increasing the number of records in each fold. However, the missingness of the data sets (proportion of incomplete to complete records) will be the ultimate determinate of how likely such scenarios were to occur. By introducing an error tolerance, indicated in *blue* for those results that varied by less than ±5.0%, we can see that the only metric where the other techniques produced better results than our algorithm was *Correlation*; the results for *TPR* and *Recall* changed marginally, and the other metrics showed good improvements achieved by our algorithm. Applying the MAV and MDAV imputation techniques shows very similar results, which may have been caused by the relatively sparse data sets, the size of the data sets, or the nature of the data itself.

As shown in Table 4, the results differed depending on which imputation method was used. When incomplete records were removed as part of the imputation process prior to the application of our N-Similarity algorithm, all of the metrics, apart from accuracy, were worsened, albeit on a restricted data set. The results of using either the mean or modal replacement approaches were very similar and could be due to the relatively small data sets used in our tests. What can be taken from this is the importance of fine tuning expectations based on which metrics are the most important to the end user. Considering our neighborhood-similarity-based approach (Table 4), we obtained better results for accuracy (+9.33%), precision (+9.67%), specificity (+13.84%), and FPR (13.86%), but this has to be tempered against worse results for correlation (−6.07%). The recall and TPR were roughly unchanged and remained within a 5% tolerance. What has become apparent is that the metrics used are very susceptible to the neighborhood size (N) and nature of the data to which they are being applied. The best results may be achieved by balancing the size of the neighborhood considered against the imputation algorithm that will be used to identify the most suitable compromise between true positive and true negative outcomes. In our testing, we ran our algorithm using different-sized neighborhoods and found that a neighborhood of size four ($N = 4$) gave the most balanced results. For comparison, the results obtained using other neighborhood sizes can be seen in Table 1.

Table 4. Performance of our proposed N-Similarity algorithm compared against other single imputation techniques.

	Remove Incomplete Records (Listwise Deletion)	**Replace Missing Data with MAV**	**Replace Missing Data with MDAV**	**Average N-Similarity Algorithm (N = 1. . . 10)**
Number Of Perfect Tests	10	10	10	10
Accuracy	54.76%	54.85%	54.88%	64.16% (+9.33%)
Correlation	92.48%	94.92%	95.00%	88.06% (−6.07%)
Precision	36.94%	31.31%	31.32%	42.86% (+9.67%)
Recall	31.26%	36.65%	37.35%	32.06% (−3.03%)
Specificity	68.96%	63.28%	62.82%	78.86% (+13.84%)
True Positive Rate (TPR)	22.23%	25.17%	25.21%	22.47% (−1.73%)
False Positive Rate (FPR)	31.04%	36.72%	37.18%	21.12% (−13.86%)
Average MCC	**0.0891**	**0.0160**	**−0.0413**	

4.4.3. Benchmarking with kNN

Using the PIMA dataset (Table 3), we also compared our similarity-based imputation (NSIM), its enhanced version NSIM-EB (with the empirical Bayes (EB) correction, $\lambda = 1$), and the Mahalanobis distance based k-nearest neighbors (kNNs) [34] (See Table 5 for the obtained RMSEs). Both the kNNs imputation and NSIM are nonparametric and rely on the k-nearest and N-similar observations, respectively. Apart from the use of neighborhood observations, the Mahalanobis distance uses the covariance matrix, while our EB correction (3) uses two estimates of sample variance as the weight of the imputed proposal. We used the three schemes (i.e., NSIM, NSIM-EB, and kNNs) for 1000 imputations per variable—for a total of M combinations (Table 5). As seen in Table 5, the RMSE performance of the NSIM was comparable to the kNN imputation, whereas the NSIM-EB outperformed both in all scenarios with minimal computational time overhead (the NSIM, NSIM-EB, and kNNs took approximately 68, 103, and 477 s, respectively, in our R implementation).

Table 5. Imputation RMSEs for simulated data using our similarity method (NSIM), our similarity method with empirical Bayes correction (NSIM-EB), and the k-nearest neighbors (kNNs) method.

M	**Method**	**Pregnancy**	**Glucose**	**BP**	**Triceps**	**Insulin**	**BMI**	**DPf**	**Age**
1	NSIM	0.875	0.963	1.051	0.937	0.942	0.892	1.103	0.848
	NSIM-EB	0.737	0.770	0.777	0.816	0.726	0.782	0.791	0.752
	kNNs	0.872	0.948	1.101	1.043	0.896	0.968	1.013	0.884
5	NSIM	1.134	1.114	1.275	1.128	1.148	1.065	1.288	1.089
	NSIM-EB	0.900	0.899	0.962	0.948	0.882	0.899	0.937	0.894
	kNNs	1.343	1.328	1.265	1.315	1.230	1.322	1.272	1.289
10	NSIM	1.172	1.149	1.335	1.167	1.235	1.096	1.372	1.125
	NSIM-EB	0.942	0.928	0.992	0.958	0.956	0.927	0.984	0.891
	kNNs	1.382	1.356	1.331	1.406	1.293	1.360	1.349	1.263
15	NSIM	1.177	11.54	1.350	1.181	1.249	1.109	1.388	1.140
	NSIM-EB	0.944	0.940	1.004	0.971	0.961	0.936	1.030	0.917
	kNNs	1.419	1.370	1.376	1.418	1.333	1.359	1.404	1.379
20	NSIM	1.187	1.167	1.356	1.185	1.269	1.121	1.393	1.160
	NSIM-EB	0.959	0.942	1.006	0.969	0.995	0.950	1.014	0.928
	kNNs	1.399	1.359	1.378	1.397	1.336	1.367	1.345	1.372

Our algorithm performed better when the source data set had a small percentage of missing data values, due to our blind random selection of data values across all the folds. The larger the number of missing data values, the higher the likelihood would be that some of the folds would be more sparsely populated. The choice of the number of data partitions in the k-fold step needs to be carefully selected; otherwise, we risk the possibility of introducing bias into the selection of data values put in any given fold. We settled on k = 10, as much of the academic literature indicated that this was a commonly used value. One way of limiting the impact of this problem is to use a stratified approach as mentioned above. We left this direction as a line of potential future work. The choice of the size of the neighborhood, *N*, and, as a direct result, the number of candidates in the set of values for selecting imputed values, was also sensitive. We spent considerable trial-and-error effort looking for the best selection for this parameter against the PIMA data set; we tried a much wider range of potential values for *N* than are shown. The results for these higher values were negligibly different in our case.

Table 1 shows that N = 4 was the best choice in our case, although this could vary for different data sets. Further research is required to determine whether the choice of value for *N* could be automated by looking at all the possible potential values for *N* and whether this approach would even be practical for large data sets in terms of processing time and improvements in the results.

5. Conclusions

Our neighborhood-based algorithm was able to provide noticeably improved results when compared against other techniques, but the degree of this improvement was sensitive to the size of the neighborhood, with some features being more readily improved than others for smaller neighborhood sizes and other metrics being noticeably less well predicted as the size of the neighborhood increased. This paper proposes a technique to provide a more accurate prognosis of possible patient diabetes based on a number of key patient characteristics. Our approach creates a similarity neighborhood using the most similar diagnosed patient records and uses the feature set values of these patients to help with the diagnosis of undiagnosed patients. By comparing our N-Similarity algorithm against several widely used single-pass imputation techniques using the same collection of data sets, both real-world and simulated, we found that it produces better results against several of our performance metrics (Table 4). However, we observed that the size of the neighborhood had an impact on the performance of our algorithm. We also noticed that the limited data set sizes and degrees of missingness of the initial source data could impact the results, and more extensive work would be necessary using a wider range of different data sets in order to see how these measures are related. The empirical Bayes correction of the neighborhood-based algorithm offered consistently smaller RMSEs over the simple algorithm and the k-nearest neighbors imputation, with minimal computational overhead. In addition to the performance advantages, we recommended it as a general method, since the shrinking parameter α indicated a degree of certainty between our inputted value and the sample mean (with zero indicating certainty of the similarity of the inputted value and one indicating most uncertain).

6. Future Work

The main limitation of our current work is that the PIMA data set contains only numeric feature values. Future work could include support for both categorical and textual data. Both types of information are widely found in medical data sets and would help to support the usefulness of our algorithm in this domain, as well as in other similar domains. The implementation of our algorithm has been deliberately developed to be loosely coupled to the source data to allow for different file formats and structures in the source data to be supported with minimal effort, thus allowing for generalization of the code for different future uses.

To aid with future development of this algorithm, we have provided the full source code to the software we used to generate the presented results. The source code, written in the Go programming language, can be freely used and modified, and it has been designed to be modular and loosely coupled to any data set, thereby making it easier to extend as required.

Author Contributions: Conceptualization, C.W., V.G. and S.D.; Methodology, C.W., V.G. and S.D.; Software, C.W.; Formal analysis, V.G.; Investigation, C.W.; Data curation, C.W.; Writing—original draft, C.W.; Writing—review & editing, V.G. and S.D.; Supervision, V.G. and S.D.; Project administration, S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The main Github repository can be found here: https://github.com/ColinWilcox1967/PhD-DataSetAnalysis-Diabetes (accessed on 22 November 2023); An example of how this code may be used with other data sets is given here: https://github.com/ColinWilcox1967/PHD-DataSetAnalysis-Traffic (accessed on 22 November 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RMSE	**R**oot **M**ean **S**quared **E**rror
NSIM	Our **N**eighbourhood **SIM**ilarity algorithm
NSIM-EB	Our **N**eighbourhood **SIM**ilarity algorithm with **E**mpirical **B**ayes Correction
kNN	Classification of **N**earest **N**eighbour algorithms
TP	**T**rue **P**ositive
FP	**F**alse **P**ositive
TN	**T**rue **N**egative
FN	**F**alse **N**egative
TPR	**T**rue **P**ositive **R**ate
FPR	**F**alse **P**ositive **R**ate
MAV	**M**ean **A**verage **V**alue
MDAV	**M**o**d**al **A**verage **V**alue
MCC	**M**atthews **C**orrelation
BMI	**B**ody **M**ass **I**ndex
BP	**B**lood **P**ressure
DPf	**D**iabetes **P**edigree **F**unction

References

1. Tang, J.; Zhang, X.; Yin, W.; Zou, Y.; Wang, Y. Missing data imputation for traffic flow based on combination of fuzzy neural network and rough set theory. *J. Intell. Transp. Syst. Technol. Plan. Oper.* **2019**, *5*, 439–454. [CrossRef]
2. Agrawal, R.; Prabakaran, S. Big data in digital healthcare: Lessons learnt and recommendations for general practice. *Heredity* **2020**, *124*, 525–534. [CrossRef] [PubMed]
3. Adam, K. Big Data Analysis And Storage. In Proceedings of the 2015 International Conference on Operations Excellence and Service Engineering, Orlando, FL, USA, 10–11 September 2015; pp. 648–658.
4. Ford, E.; Rooney, P.; Hurley, P.; Oliver, S.; Bremner, S.; Cassell, J. Can the Use of Bayesian Analysis Methods Correct for Incompleteness in Electronic Health Records Diagnosis Data? Development of a Novel Method Using Simulated and Real-Life Clinical Data. *Public Health* **2020**, 8, 54. [CrossRef] [PubMed]
5. Xiaochen, L.; Xia, W.; Liyong, Z.; Wei, L. Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. *Neurocomputing* **2019**, *266*, 54–65. [CrossRef]
6. Singhal, S. Defining, Analysing, and Implementing Imputation Techniques. 2021. Available online: https://www.analyticsvidhya.com/blog/2021/06/defining-analysing-and-implementing-imputation-techniques/ (accessed on 22 November 2023).
7. Beretta, L.; Santaniello, A. Nearest neighbor imputation algorithms: A critical evaluation. *BMC Med. Inform. Decis. Mak.* **2016**, *16*, 197–208. [CrossRef] [PubMed]
8. Khaled, F.; Mahmoud, I.; Ahmad, A.; Arafa, M. Advanced methods for missing values imputation based on similarity learning. *Clim. Res.* **2022**, *7*, e619. [CrossRef]
9. Huang, G. Missing data filling method based on linear interpolation and lightgbm. *J. Phys. Conf. Ser.* **2021**. [CrossRef]
10. Peppanen, J.; Zhang, X.; Grijalva, S.; Reno, M.J. Handling bad or missing smart meter data through advanced data imputation. In Proceedings of the 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Ljubljana, Slovenia, 9–12 October 2016; pp. 1–5. [CrossRef]
11. Jackobsen, J.; Gluud, C.; Wetterslev, J.; Winkel, P. When and how should multiple imputation be used for handling missing data in randomised clinical trials—A practical guide with flowcharts. *BMC Med. Res. Methodol.* **2017**, *17*, 162. [CrossRef]
12. Hayati Rezvan, P., Lee, K.J.; Simpson, J.A. The rise of multiple imputation: A review of the reporting and implementation of the method in medical research. *BMC Med. Res. Methodol.* **2015**, *15*, 30. [CrossRef]

13. Nguyen, C.; Carlin, J.; Lee, K. Practical strategies for handling breakdown of multiple imputation procedures. *Emergent Themes Epidemiol.* **2021**, *18*, 5. [CrossRef]
14. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN Model-Based Approach in Classification. In *Confederated International Conferences "On The Move To Meaningful Internet Systems 2003"*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2888, pp. 986–996. [CrossRef]
15. Pohl, S.; Becker, B. Performance of Missing Data Approaches Under Nonignorable Missing Data Conditions. *Methodology* **2018**, *16*, 147–165. [CrossRef]
16. Ali, N.; Neagu, D.; Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SN Appl. Sci.* **2019**, *1*, 1559. [CrossRef]
17. Abu Alfeilat, H.A.; Hassanat, A.B.A.; Lasassmeh, O.; Tarawneh, A.S.; Alhasanat, M.B.; Eyal Salman, H.S.; Prasath, V.S. Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data* **2019**, *7*, 221–248. [CrossRef]
18. Khan, S.; Hoque, A. SICE: An improved missing data imputation technique. *J. Big Data* **2020**, *7*, 37. Available online: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00313-w (accessed on 22 November 2023). [CrossRef] [PubMed]
19. Misztal, M. Imputation of Missing Data Using R. *Acta Univ. Lodz. Folia Oeconomica* **2012**, *269*, 131–144.
20. Kowarik, A.; Templ, M. Imputation with the R Package VIM. *J. Stat. Softw.* **2016**, *74*, 1–16. [CrossRef]
21. Choi, J.; Dekkers, O.; Le Cessie, S. A comparison of different methods to handle missing data in the context of propensity score analysis. *Eur. J. Epidemiol.* **2019**, *34*, 23–36. [CrossRef]
22. Cetin-Berber, D.; Sari, H. Imputation Methods to Deal With Missing Responses in Computerized Adaptive Multistage Testing. *Educ. Psychol. Meas.* **2018**, *79*, 495–511. [CrossRef]
23. Alwohaibi, M.; Alzaqebah, M. A hybrid multi-stage learning technique based on brain storming optimization algorithm for breast cancer recurrence prediction. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *34*, 5192–5203. [CrossRef]
24. Kabir, G.; Tesfamariam, S.; Hemsing, J.; Rehan, S. Handling incomplete and missing data in water network database using imputation methods. *Sustain. Resilient Infrastruct.* **2020**, *5*, 365–377. [CrossRef]
25. Mujahid, M.; Rustam, F.; Shafique, R.; Chunduri, V.; Villar, M.G.; Ballester, J.B.; Diez, I.D.L.T.; Ashraf, I. Analyzing Sentiments Regarding ChatGPT Using Novel BERT: A Machine Learning Approach. *Information* **2023**, *14*, 474.
26. Mujahid, M.; Rehman, A.; Alam, T.; Alamri, F.S.; Fati, S.M.; Saba, T. An Efficient Ensemble Approach for Alzheimer's Disease Detection Using an Adaptive Synthetic Technique and Deep Learning. *Diagnostics* **2023**, *13*, 2489. [CrossRef] [PubMed]
27. Nti, I.; Nyarko-Boateng, O.; Aning, J. *Performance of Machine Learning Algorithms with Different K Values in K-Fold Cross Validation*; MECS Press: Hong Kong, China, 2021. [CrossRef]
28. Brownlee, J. *How to Configure k-Fold Cross-Validation*; Machine Learning Mastery: San Juan, PR, USA, 2020.
29. Little, R.J.; Rubin, D.B. *Statistical Analysis with Missing Data*; John Wiley & Sons: Hoboken, NJ, USA, 2019; Volume 793.
30. Carlin, B.; Louis, T. *Bayes and Empirical Bayes Methods for Data Analysis*, 2nd ed.; Chapman and Hall CRC: Boca Raton, FL, USA, 2000. [CrossRef]
31. Zhou, X.; Wang, X.; Dougherty, E.R. Missing-value estimation using linear and non-linear regression with Bayesian gene selection. *Bioinformatics* **2003**, *19*, 2302–2307. [CrossRef] [PubMed]
32. Cheng, P.E. Nonparametric Estimation of Mean Functionals with Data Missing at Random. *J. Am. Stat. Assoc.* **1994**, *89*, 81–87. [CrossRef]
33. Root Mean Squared Error Definition. 2022. Available online: https://www.sciencedirect.com/topics/engineering/root-mean-squared-error (accessed on 22 November 2023).
34. Crookston, N.L.; Finley, A.O. yaImpute: An R package for kNN imputation. *J. Stat. Softw.* **2008**, *23*, 1–16. [CrossRef]
35. PIMA Indian Diabetes Database. 2016. Available online: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database (accessed on 22 November 2023).
36. Lin, W.C.; Chih-Fong, T. Missing value imputation: A review and analysis of the literature (2006–2017). *Artif. Intell. Rev.* **2020**, *53*, 1487–1509. [CrossRef]
37. Dong Y, P.C. Principled missing data methods for researchers. *Springerplus* **2013**, *2*, 222. [CrossRef]
38. Huang, L.; Wang, C.; Rosenberg, N.A. The Relationship between Imputation Error and Statistical Power in Genetic Association Studies in Diverse Populations. *Am. J. Hum. Genet.* **2009**, *85*, 692–698. [CrossRef]
39. Pepinsky, T.B. A Note on Listwise Deletion versus Multiple Imputation. *Political Anal.* **2018**, *26*, 480–488. [CrossRef]
40. Lall, R. How multiple imputation makes a difference. *Political Anal.* **2006**, *24*, 414–433. [CrossRef]
41. Allison, P. Listwise Deletion: It's NOT Evil. 2014. Available online: https://statisticalhorizons.com/listwise-deletion-its-not-evil/ (accessed on 22 November 2023).
42. Joachim Schork, S.G. Imputation Methods (Top 5 Popularity Ranking). 2019. Available online: https://statisticsglobe.com/imputation-methods-for-handling-missing-data/ (accessed on 22 November 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.