Journal of Food Composition and Analysis 112 (2022) 104638

Contents lists available at ScienceDirect

Journal of Food Composition and Analysis

journal homepage: www.elsevier.com/locate/jfca

Missing value imputation in food composition data with denoising autoencoders

Ivana Gjorshoska [a,*], Tome Eftimov [b], Dimitar Trajanov [a,c]

[a] Faculty of Computer Science and Engineering, "Ss. Cyril and Methodius" University-Skopje, ul. Rudzer Boshkovikj 16, P.O. 393, 1000 Skopje, Republic of North Macedonia
[b] Computer Systems Department, Jožef Stefan Institute, Jamova Cesta 39, 1000 Ljubljana, Slovenia
[c] Department of Computer Science, Metropolitan College, Boston University, Boston, USA

ARTICLE INFO

Keywords:
Food composition data
Food composition databases
Nutrient values
Missing data
Missing value imputation
Autoencoders
Deep learning

ABSTRACT

Missing data is a common problem in a wide range of fields that can arise as a result of different reasons: lack of analysis, mishandling samples, measurement error, etc. The area of nutrition and food composition is no exception to the problem of missing values. Missing data in food composition databases (FCDB) significantly limits their usage. Commonly this problem is resolved by calculating mean or median from available data in the same FCDB or borrowing values from other FCDBs, however, this method produces notable errors. This paper focuses on missing value imputation using autoencoders, a deep learning algorithm that has the ability to approximate values by learning a higher-level representation of its input. The data used was from the FCDBs collected by the USDA FoodData Central. We compared the autoencoder imputation method with the commonly used approaches fill-in-with-mean and fill-in-with-median, and the results show that the autoencoder method for imputation provides superior results.

1. Introduction

Food Composition Data (FCD) are detailed sets that provide information about the nutritional components of food, including values for nutrients, energy, and other bioactive components, as well as metadata such as classifiers and descriptors (Greenfield and Southgate, 2003). These sets are structured and available in Food Composition Databases (FCDB), which are the primary data sources used in Food and Nutrition Science as well as in other public-health domains, food industry, and clinical practice. FCD is also widely used in the assessment of nutrient intake at an individual, regional, national, and international level (Williamson, 2006).

Due to the wide variety of sources and the ways in which the data is obtained, the FCD contained in FCDBs varies in quality. There are codes and references for the data types and sources to identify the data with a specific order of preference (EuroFIR, 2021; Agricultural Research Service, 2021; Health Canada and Branch, 2018). The most preferred way of obtaining data for FCDBs is to use original analytic values for the food composition data, i.e., data taken from published literature or laboratory reports. Other alternatives for obtaining data are using estimated data derived from analytical values obtained from a similar food or another form of the food, data derived from recipes and calculated from the nutrient contents of the ingredients, and lastly, borrowed values from other tables or databases.

The quantity of data and its elements, as well as the metadata describing them, differs from one database to another. Because of this, there are several limitations that constrict their usage (Ispirova et al., 2020). Some of these limitations include the incompatibility of the databases, limited and incomplete coverage of food items, limited and incomplete coverage of nutrients, errors when using the database, inadequate database or values, and limitations of methods to measure food intake. One of the biggest limitations of these databases that significantly restricts their use is the incomplete coverage of foods and nutrients, leading to missing data. With the increasing growth of food demand and supply, relying only on chemical analysis for FCDBs is almost impossible. There are a few ways to deal with missing data, and three of the most common methods are ignoring the missing data, i.e., omitting the instances that contain missing data, imputing plausible estimated values in place of the missing values, or using model-based techniques, i.e., defining a model from the available data based on its distribution. Ignoring strategies are usually the default method for dealing with missing data, and they are used in cases with a minimal

* Corresponding author.
E-mail addresses: gjorshoskaivana@gmail.com (I. Gjorshoska), tome.eftimov@ijs.si (T. Eftimov), dimitar.trajanov@finki.ukim.mk (D. Trajanov).

https://doi.org/10.1016/j.jfca.2022.104638
Received 2 November 2021; Received in revised form 26 March 2022; Accepted 13 May 2022
Available online 16 May 2022
0889-1575/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Table 1
Sample data from "food nutrient" table.

Food ID	Nutrient ID	Amount	Data points	Derivation ID	Min	Max	Median	Footnote	Year acquired
319877	1051	56.30	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319877	1002	1.28	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319877	1004	19.00	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319877	1007	1.98	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319878	1091	188.00	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319878	1101	1.21	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319878	1092	326.00	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319878	1087	46.00	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319878	1093	446.00	1.0	1.0	NaN	NaN	NaN	NaN	NaN
319878	1090	75.90	1.0	1.0	NaN	NaN	NaN	NaN	NaN

amount of missing data. For now, one of the most used approaches to resolve the missing data problem in FCDBs is to borrow data from databases of other countries or calculating mean or median values from similar foods from the same FCDB (Greenfield, 1995). Furthermore, if borrowing data is done to construct a national database, there are suggestions that should be applied. These suggestions state that missing values should be borrowed from FCDBs, which contain the foods and nutrients of interest as well as being from geographically similar countries (Church, 2009). Frequently, imputation of missing data by borrowing is done in one of two ways: using one database well known for its quality, or using several databases and calculating either arithmetic mean or median of the respective values (Forrest et al., 2013). These approaches are generally inaccurate due to the fact that the composition of any given food sample cannot be accurately predicted and lead to lowering the quality of the FCBDs. Consequently, there is a clear need for better methods that can be used for the imputation and calculation of missing FCD.

In this paper, we propose using autoencoders as a state-of-the-art method to impute missing values in FCDBs. Advancements in deep learning have resulted in architectures that have the capacity to learn complex representations, which is not possible using classical models. We present 2 methods of calculating missing values. The first approach is the traditional approach for borrowing FCD - calculating mean and median from existing data, while the second approach involves using Denoising Autoencoders (DAEs) (Vincent et al., 2008). These models are designed to recover clean output from noisy input, and since missing data is a special case of noisy input, DAEs are an ideal imputation model candidate. We propose a multiple imputation framework with DAE as the base model, and through several experiments, we demonstrate that our methodology provides higher accuracy when compared to the commonly used approaches.

The rest of the paper is organized as follows. In Section 2, we describe the data used in the experiments and the methods used to calculate the missing values. Section 3 describes the autoencoder architecture and the evaluation criteria. In Section 4, we present the experiments in detail, compare the results obtained by each method and demonstrate that the use of autoencoders provides higher accuracy than the traditional approaches. In Section 5 we discuss possible advantages of using autoencoders as an imputation model as well as the challenges we faced during the experiments, and finally, Section 6 provides a summary of the importance of the methods explained in this paper and directions for future work.

2. Methods and materials

This section explains the data used in our experiments, how it is obtained and structured and the format in which it is used, followed by an explanation of the imputation methods, both the traditional approach and the deep learning imputation method using autoencoders.

Missing values are unobserved values in a dataset and they can be missing for different reasons. According to the mechanism of missingness, there are three types of missing data that can occur and FCDBs contain missing values of each type. These types describe the relationship between the probability of a missing value and the other variables in the dataset.

1. Missing at random (MAR) - The probability that a value is missing depends only on the observed values X and not on the unobserved values Y (Eq. (1)). The missing data is just a random subset of the data and the reason for it's missingness can be explained using observed data. In FCD, this type of missing data can be found if, for example, an analysis has been conducted for certain nutrients in certain foods and the data is entered, but in the database there are nutrients that have not been observed with the analysis. This creates missing values in FCDBs for the values of the unobserved nutrients for the same foods.

$$\mathrm{P(Yismissing{=}X,Y)} = \mathrm{P(Yismissing{=}X)} \tag{1}$$

2. Missing completely at random (MCAR) - The probability of missingness is not dependent on any observed X or unobserved Y values (Eq. (2)). One example of MCAR could be a malfunction when entering the data resulting in deletion of some of the data values.

$$P(Yismissing{=}X,Y) = P(Yismissing) \tag{2}$$

3. Missing not at random (MNAR) - Occurs when the conditions of MAR are violated, meaning the probability of missingness is dependent on the unobserved attributes (Y in Eq. (3)) or on the missing attribute itself. In FCDBs, this type of missing data might happen for foods that are not cultivated in the country the FCDB originates from because of the unfavorable climate. In this case, the weather conditions is the unobserved variable that leads to missingness.

$$\mathrm{P(Yismissing{=}X,Y)} = \mathrm{P(Yismissing{=}Y)} \tag{3}$$

2.1. Food composition data

Currently, there are several international organizations and projects that work with food composition. One of them is the United States Department of Agriculture (USDA) which is a federal department responsible for developing and executing federal laws related to food, farming, forestry and rural economic development (USDA, 2021). The department consists of 29 agencies, one of which is the Agricultural Research Service (ARS), the in-house research agency for the USDA and its main job is to find solutions for agricultural problems affecting the foods production and consumption. For the purpose of its research projects, the ARS collects and maintains data for nutrition and food safety, crop and animal production and protection and natural resources. The agency maintains the FoodData Central (Agricultural

Table 2
Format of the newly constructed datasets.

Food ID	1002	1003	1004	1005	1007	1008	1009	1010	1011	1012
319883	1.28	0	19	0	1.98	0	0	0	0	0
319906	0	0	0	0	0	0	0	0	0	0
319907	1.29	0	18.7	0	1.99	0	0	0	0	0
319909	0	0	0	0	0	0	0	0	0	0
319910	0	0	0	0	0	0	0	0	0	0
319918	1.25	0	16.6	0	2.02	0	0	0	0	0
319919	0	0	0	0	0	0	0	0	0	0
319920	1.1	0	19.1	0	1.98	0	0	0	0	0
319926	0	0	0	0	0	0	0	0	0	0
319927	0	0	0	0	0	0	0	0	0	0
319929	0	0	0	0	0	0	8.2	0	0	0
319933	1.28	0	18.2	0	1.99	0	0	0	0	0

Table 3
Sample data from "food" table.

Food ID	Data type	Description	Food category ID	Publication date
319874	sample food	HUMMUS, SABRA CLASSIC	16	2019–04–01
320082	sub sample food	Milk, 2%	1	2019–04–01
320358	market acquisition	Beef, eye of round roast, raw (ER37-R-10)	13	2019–04–01
319877	sub sample food	Hummus	16	2019–04–01
320413	sample food	TOMATOES, GRAPE	11	2019–04–01
321363	sub sample food	Salt; iodized	2	2019–04–01
321513	market acquisition	BEANS, SNAP, CANNED, DRAINED, DEL MONTE	11	2019–04–01
321721	market acquisition	Broccoli, Region 4, PA4, NFY0104OG	11	2019–04–01
323539	sub sample food	Whole eggs	1	2019–04–01

Research Service, 2021), an integrated data system that provides expanded nutrient profile data and links to related agricultural and experimental research. There are five distinct types of data that provide information on nutrients and other food components.

In our experiments, we used the Foundation Foods data type which includes values derived from analyses for food components, including a wide variety of nutrients and extensive metadata, such as number of samples taken into account, analytical approaches used, sampling location, etc. The dataset of interest is the table annotated as "food nutrient" which contains data for a nutrient value in a given food and its structure is shown in Table 1. Every entry in the table consists of several elements: food ID to represent a food item, nutrient ID to which the food nutrient pertains to, the amount of the nutrient per 100 g of food specified in the unit defined for the nutrient, the number of observations and the technique used to derive the values, the minimum, maximum and median values for the amount of the nutrient in the food etc. We constructed two new datasets, one using the average amount data and one using the median amount data. The newly constructed datasets have 13,105 rows and 198 columns each and portray each food as a vector of values where each value represents the amount of a specific nutrient in the food i.e every column in the dataset is represented by an ID of a nutrient and every sample represents one food item, as shown in Table 2. Zero values were assigned to the food-nutrient pairs that do not exist in the original dataset. In the original dataset "food nutrient", the column containing the average values has no missing values whereas the column portraying the median values contains a lot of missing values, namely out of 97,985 samples, only 9620 samples contain a value for the median amount of a specific nutrient in a specific food item. As a consequence, the previously mentioned restructuring of the datasets resulted in the final datasets being populated with a lot of zero values.

2.1.1. *Word embeddings*

Word embeddings are a commonly used method in Natural Language Processing (NLP) that encodes the meaning of each word such that the words that are closer to each other are similar in meaning (Martin, 2009). Conceptually, it involves mathematical embedding from a space with a lot of dimensions per word to a continuous vector space with a constant dimension. This technique represents individual words as real-valued vectors in a lower dimensional space that preserves the semantic relationships of the words. Word embeddings can be generated using several methods such as neural networks (Mikolov et al., 2013), probabilistic methods (Globerson et al., 2007) and dimensionality reduction of the co-occurrence matrix (Li et al., 2015; Lebret and Collobert, 2013).

Based on the fact that word embeddings showcase the semantic relationship between similar words, food items with similar names will have similar embedding vectors and consequently, these foods should have similar nutrient composition vectors. To test this hypothesis, we used the food descriptions that can be found in the "food" dataset (Table 3) of the Foundation Foods data type (Agricultural Research Service, 2021) to convert the description into embeddings and use them during the autoencoder training and testing phases. We used only the food descriptions for each food item to get a vector mapping. We attached these word embedding vectors as additional columns in each dataset and used them as latent variables to indicate that food items with similar values in their word embeddings should have similar values for the respective nutrients.

2.2. *Related work*

In this section, we present the related work needed to understand the methodologies used in our experiments. We introduce the traditional and statistical approaches for missing value imputation in FCD.

2.2.1. *Traditional approaches for borrowing FCD*

When using FCDBs, there are several considerations the users need to take into account in order to ensure that the values the FCDB provides are used correctly. Some of these considerations include variability of the composition of a food item, misunderstanding of nutrient definitions, use of an incorrect conversion factor and use of nutrient interchangeably with its sub-types (Ispirova et al., 2020). There are also limitations that should be considered when using FCDBs such as variability in the composition of a food item, the number of food items and range of nutrients covered by a database.

Nowadays, a lot of countries have their own national FCDBs available online (EuroFIR, 2021; Agricultural Research Service, 2021). These databases differ from one another because of many factors, such as different cultivators, soil, climates and agricultural practices, differing food production and processing practices, variation in recipe ingredients composition and variation in available food products. Varied nutrient

definitions and values, analytical methods used to obtain the data, nutrient calculations and quantity of foods contained in the database result in difference in FCD between countries. There can also be differences in standard measure for food (per 100 g measure has been outlined as the standard) and the way missing values are treated. Because of these differences, there are several factors that need to be taken into account when borrowing FCD from another FCDB. The chosen FCDB should:

1. Contain foods and nutrients of interest
2. Contain up-to-date FCD
3. Be of high quality i.e follow international guidelines and standards for generation of FCD: FCDBs are updated regularly, foods and components are well defined using appropriate analytical methods and nutrient definitions and the data contains variety of food types (raw, cooked, recipes, supplements etc.)
4. Come from a country that is similar in respect to geographic location, agriculture, food production and processing and recipes

The aforementioned rules are often not taken into account because they are hard to follow due to the fact that rarely are all four of them fulfilled. As a result, the users of FCDBs choose other ways to find a suitable FCDB to borrow values from. The most common approach is to borrow from a FCDB of a neighboring country or from a FCDB with a large span of data, however this solution may lead to a variety of errors. Consequently, mean and median calculations are easier and more reliable choice for dealing with missing data in FCDBs.

In our experiments, we used the previously mentioned traditional methods for missing value imputation - imputation with mean and median values. We tried two approaches for comparison. The first approach is by calculating and imputing the mean or median value for each nutrient across all foods in the dataset and the second approach for imputation is to calculate and impute the mean or median value for a certain nutrient of similar foods in the same FCDB i.e. imputing based on the category the food item belongs to.

2.2.2. *Statistical approaches for missing value imputation in FCD*

Missing data in FCDBs has been a persistent problem and there has been research done in the area to improve it. Several methods for estimating nutrient values have been discussed as common approaches for solving this issue, namely using values from a different but similar food, calculating values from different forms of the same food or from other components in the same food, calculating values from recipes or commercial product formulations, calculating values from a product standard and assuming a zero value (Schakel et al., 1997). Statistical methods for handling missing data in FCDB have also been discussed using Null Hypothesis Testing (Ispirova et al., 2019). Further evaluation on statistical methods for missing values in FCD has been done in (Ispirova et al., 2020) where several methods, particularly Non-Negative Matrix Factorization (NMF), Multiple Imputations by Chained Equations (MICE), Nonparametric Missing Values Imputation using Random Forest and K-Nearest Neighbors have been evaluated and compared to filling in missing data with mean or median values. Missing data in Food Frequency Questionnaires (FFQ) (Ichikawa et al., 2019) has also been examined and several techniques for imputing missing values were tested, i.e. imputing the missing values of the same individual from a previous questionnaire, zero imputation and multiple imputation by chained equations, as well as calculating mean total energy and nutrient intake.

3. Methodology

In recent years, deep learning has become a state-of-the-art technique and has shown remarkable results in many fields. An autoencoder neural network is an unsupervised learning algorithm that tries to learn an approximation to the original input, i.e. tries to compress high-dimensional data into a lower-dimensional representation and then decompress the representation into its original form. This mapping of the original input in a different dimensional subspace enables the algorithm to learn representation for the data commonly used for dimensionality reduction as well as image compressing (Theis et al., 2017) and image denoising (Cho, 2013b; a; Gondara, 2016), machine translation (Cho et al., 2014; Sutskever et al., 2014), facial recognition (Hinton et al., 2011) etc. An autoencoder model consists of two parts - an encoder that maps the input into a different dimensional representation, and a decoder that maps the encoded representation to a reconstruction of the original input. Both the encoder and the decoder are artificial neural networks where the input layer of the encoder and the output layer of the decoder have the same number of nodes. Although these models are trained to minimize the reconstruction error using the backpropagation algorithm, they do not learn to duplicate the input, but rather to approximate it by learning only the most relevant aspects of the data. Because of this, they can be used to generate new observations of the original data and are a method to consider when opting for multiple value imputation.

Autoencoders as a deep learning algorithms have been used for the purpose of missing value imputation in various domains. (Miok et al.,

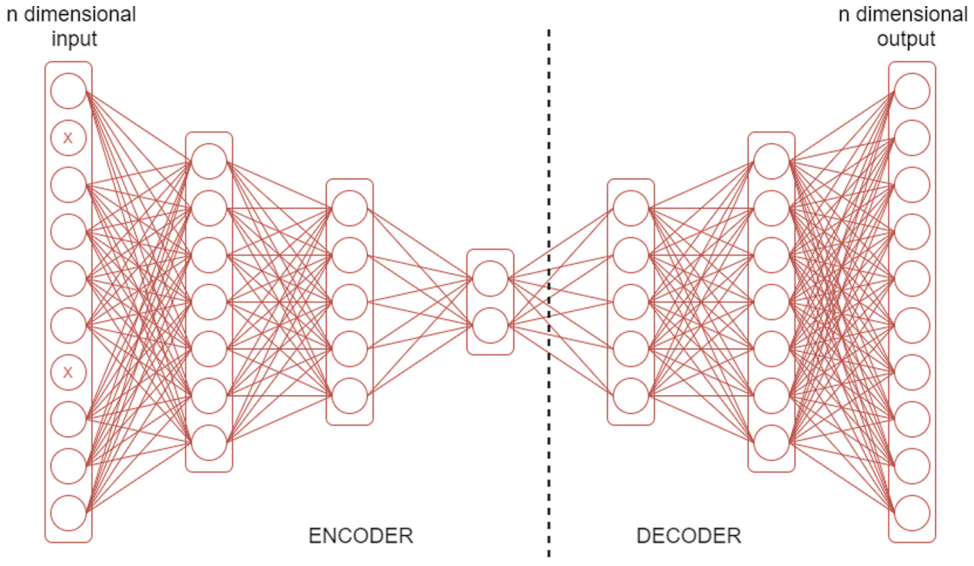


Fig. 1. Simplified autoencoder architecture.

2020) use generative autoencoders for multiple imputation in biomedical data. Variational autoencoders (VAEs), whose objective is to model the data as a distribution, have been used for imputing missing data in several previous works - (Camino et al., 2019) use VAEs as well as Generative Adversarial Networks (GAIN) for imputing missing values in tabular data sources and VAEs have also been used for imputing missing data in traffic estimation (Boquet et al., 2019). Another variation of the autoencoder models are Denoising autoencoders (DAEs) which are effective for imputing values in a given dataset by exploring non-linear correlations between the missing and the non-missing values. They have shown strong performance on a wide range of missing data problems including problems with both numerical and categorical variables (Abiri et al., 2019) as well as different data types, patterns and distributions (Gondara and Wang, 2018), estimating missing values in healthcare big data (Kim and Chung, 2020) and have been used in imputing missing values attribute-by-attribute sequentially and in one batch (Ma et al., 2020). However, prior to now, autoencoders have not been used for missing value imputation in food related data. Furthermore, there has not been a proposed method that is able to impute values for multiple different nutrients at once.

3.1. Autoencoder architecture

As mentioned previously, DAEs have been used as an imputation model for missing values in several other domains. These models are an extension to the autoencoders (Vincent et al., 2008). Their main characteristics are that they corrupt the input data and force the network reconstruct the clean output, and they force the hidden layers to learn robust features and to extract features that will establish higher level representation of the input. The training of a DAE starts with corrupting the initial input and this can be done in a few different ways such as using distributional additive noise or randomly setting some input values to zero. The corrupted input is then mapped to a hidden representation using the same process as in the standard autoencoder and from there, it is reconstructed to an approximation of the original input.

The proposed architecture is inspired by an already existing autoencoder architecture for multiple value imputation that has been tested on several different datasets and has been proven to outperform current state-of-the-art methods (Gondara and Wang, 2018). However, instead of using an overcomplete DAE representation that increases the dimensional space in each successive step in the encoder, we opted to use a more traditional approach, i.e. an autoencoder architecture that relies on dimensionality reduction in order to learn high-level characteristics of the data. Fig. 1 shows the architecture of the autoencoder model used in our experiments. The input in the encoder is the initial n dimensional vector of the food item. Then the dimensionality of the input is lowered so that 70%, 50% and 20% of the input values are taken into account in each successive layer accordingly. The percentages for dimensionality reduction used in the architecture are chosen due to the fact that these aspect ratios of dimensional reduction in relation to the dimension of the original vectors are used for information retrieval (Zhu et al., 2016; Zhuo et al., 2014). The decoder part of the autoencoder increases the hidden representation by the same previously mentioned percentages in each successive layer and outputs the final vector for the food item. To introduce corruption, there is dropout layer with a dropout ratio of 0.2, so that in a given epoch during training, 20% of the inputs are set to zero, as shown in Fig. 1. The hidden layers in the encoder and decoder neural networks use Tahn activation function. It is important to mention that this approach is based on one assumption and that is that there is enough data to train the model so it can recover the true data and not use certain values as placeholders. All of the autoencoder models used in the experiments are trained in 200 epochs using an adaptive learning rate.

Table 4
Dimensions of each dataset used for the experiments.

	Median values	Average values
Use case 1	5489 × 20	13105 × 20
	5489 × 25	13105 × 25
Use case 2	3586 × 40	13105 × 40
	3586 × 50	13105 × 50
Use case 3	1453 × 80	13105 × 80
	1453 × 100	13105 × 100
Use case 4	337 × 120	13105 × 120
	337 × 150	13105 × 150
Use case 5	/	13105 × 198
		13105 × 248

3.2. Evaluation criteria

The imputation results are compared using a frequently used measure for difference - Root mean square error. This measure presents the relative differences between the values predicted by a model or an estimator and the observed values. This value is calculated with Eq. (4). In the experiments, we calculated the root mean square error for the imputed value in each nutrient and summed them together to get the total error across all nutrients in the test set (Eq. (5)).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i^{observed} - X_i^{imputed})} \quad (4)$$

$$RMSE_{sum} = \sum_{i=1}^{m} RMSE_i \quad (5)$$

4. Results

The starting point of the experiments are the original datasets containing the average and median values. Since neural networks take only defined values as input as well as for testing purposes, we only considered the samples in the datasets that do not contain any undefined values which are all 13,105 samples for the dataset with average values and only 37 for the dataset with median values. Due to the small number of samples in the median values dataset, we chose a set of columns to perform the experiments on instead of taking into account all 198 columns. We have two groups of use cases, one for the datasets containing the median nutrient values and one for the datasets containing the average nutrient values. The use cases for each dataset are as follows:

Use Case 1: Selecting 20 nutrients from the datasets that contain the highest percentage of non zero values across all food items.

Use Case 2: Selecting 40 nutrients from the datasets that contain the highest percentage of non zero values across all food items.

Use Case 3: Selecting 80 nutrients from the datasets that contain the highest percentage of non zero values across all food items.

Use Case 4: Selecting 120 nutrients from the datasets that contain the highest percentage of non zero values across all food items.

Use Case 5: Using all 198 nutrients from the datasets - this use case was done only using the dataset containing the average nutrient values because the median values dataset contains very little entirely populated samples to perform neural network training.

For each use case, we trained two autoencoders, one using only the data of the nutrient values, and another using additional word embeddings, as mentioned in Section 2.1.1. For the embeddings, we used the description of each food item and used a sentence encoding model InferSent (Conneau et al., 2017) that provides semantic representations for English sentences. We used a pre-trained InferSent model with pre-trained GloVe word embeddings (Pennington et al., 2014) and transformed the food descriptions into vectors of 4096 values. Then, with the use of Principal Component Analysis (PCA), a process

Table 5
Average $RMSE_{sum}$ error on median datasets (20% missingness).

	20 columns	40 columns	80 columns	120 columns
Autoencoder without embeddings	**0.54**	**1.15**	**3.63**	**7.56**
Autoencoder with embeddings	0.63	1.26	3.88	7.88
Mean by category	0.80	1.48	4.09	9.79
Median by category	0.83	1.51	4.21	8.26
Mean across all foods	0.83	1.52	4.27	8.81
Median across all foods	0.83	1.51	4.21	8.26

and accordingly, the second indicate the dimensions of the datasets with word embeddings. Before conducting the experiments, we standardized the values in each dataset to be numbers between 0 and 1 in order to bring all values to a common scale since nutrients can be measured using different units. This scaling also contributes for faster convergence of the autoencoder model. Additionally, the datasets are split using a random 80–20 train-test split i.e 80% of the samples in the dataset are used for training the autoencoder and the other 20% are used for testing the methods.

In the aforementioned datasets, we introduce missing data, i.e. we set a certain percentage of the data as missing meaning we delete random data from the datasets. For the testing of the methods, we

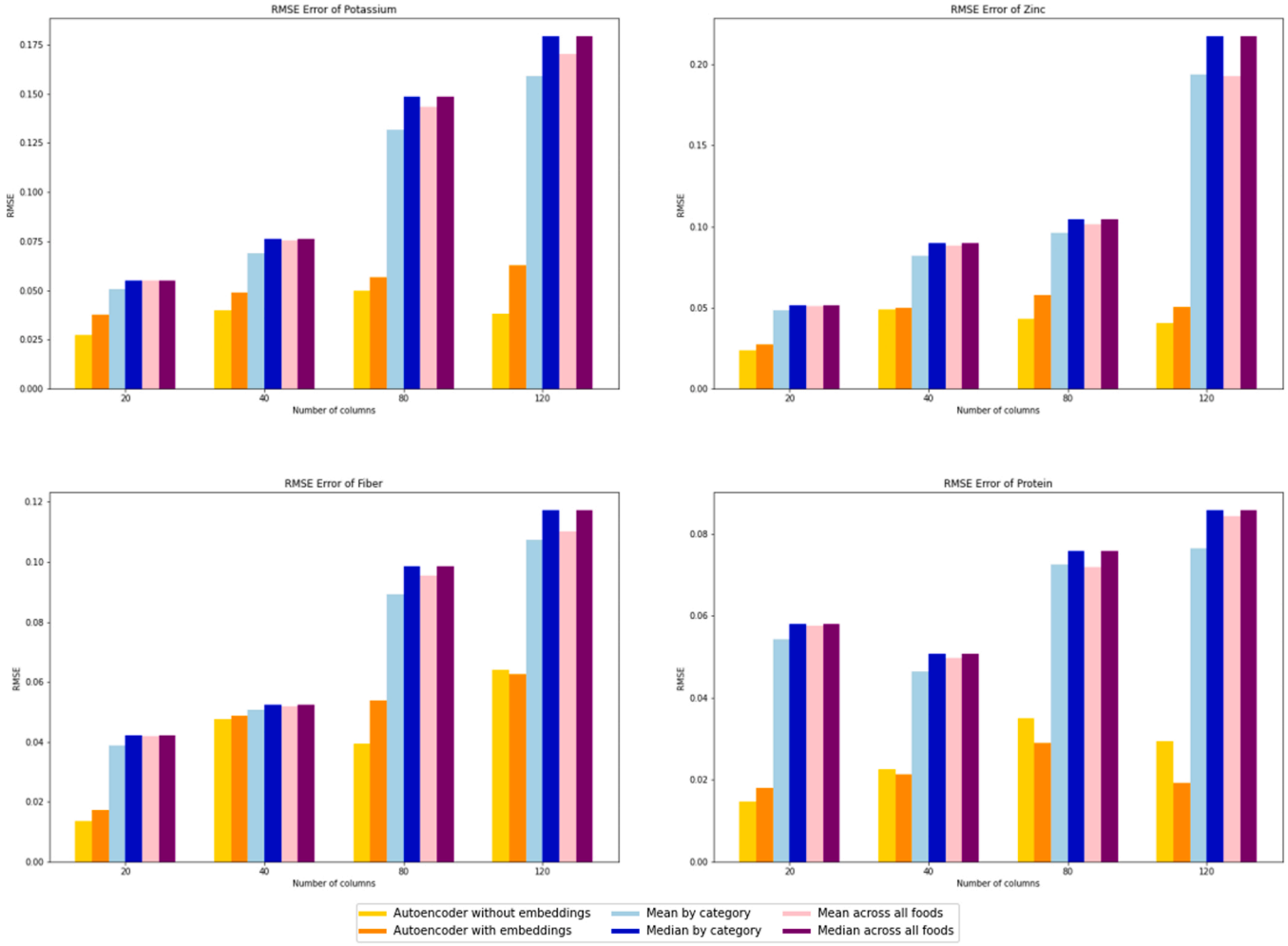


Fig. 2. RMSE results for potassium, zinc, fiber and protein in the median value datasets.

frequently used in machine learning for dimensionality reduction, we reduced the number of values in the embedding vectors so that they do not overshadow the nutrient values that the model is supposed to predict. We made the dimension of the word encodings in correlation to the number of columns in the dataset, that is, the number of values in the word encodings is roughly 25% of the number of columns in the dataset. For example, if the dataset has 20 columns representing the values of the nutrients, we add additional 5 columns that represent the encoding of the food items descriptions. Similarly, we added 10 columns representing word encodings to the datasets with 40 columns, 20 to the datasets with 80 columns, 30 to the datasets with 120 columns and 50 to the datasets with 198 columns. The dimensions of the datasets for each use case are shown in Table 4 where for each use case, the first entries indicate the dimensions of the dataset without added word embeddings

introduce MCAR missingness in the testing data with a fixed missingness proportion of 20%. This percentage of missingness introduces enough missingness and keeps enough relevant information needed to evaluate the proposed methodology. The missing data is generated by appending a random uniform vector v with n observations to the dataset with values between 0 and 1, where n is the number of samples in the dataset. Then, we set all attributes to have missing values where $v_i \leq t$, $i \epsilon 1{:}\, n$ where t is the missingness threshold of 20%. Since the autoencoder requires an input without any undefined values, the missing values in the test set used for the autoencoder are filled with the mean values for the corresponding nutrient. As an additional experiment, we tested the performance of the imputation methods for different percentages of missing data. Namely we used varying percentages of missingness from 1% to 40% with 1% increment. The maximum 40% was chosen due to the fact

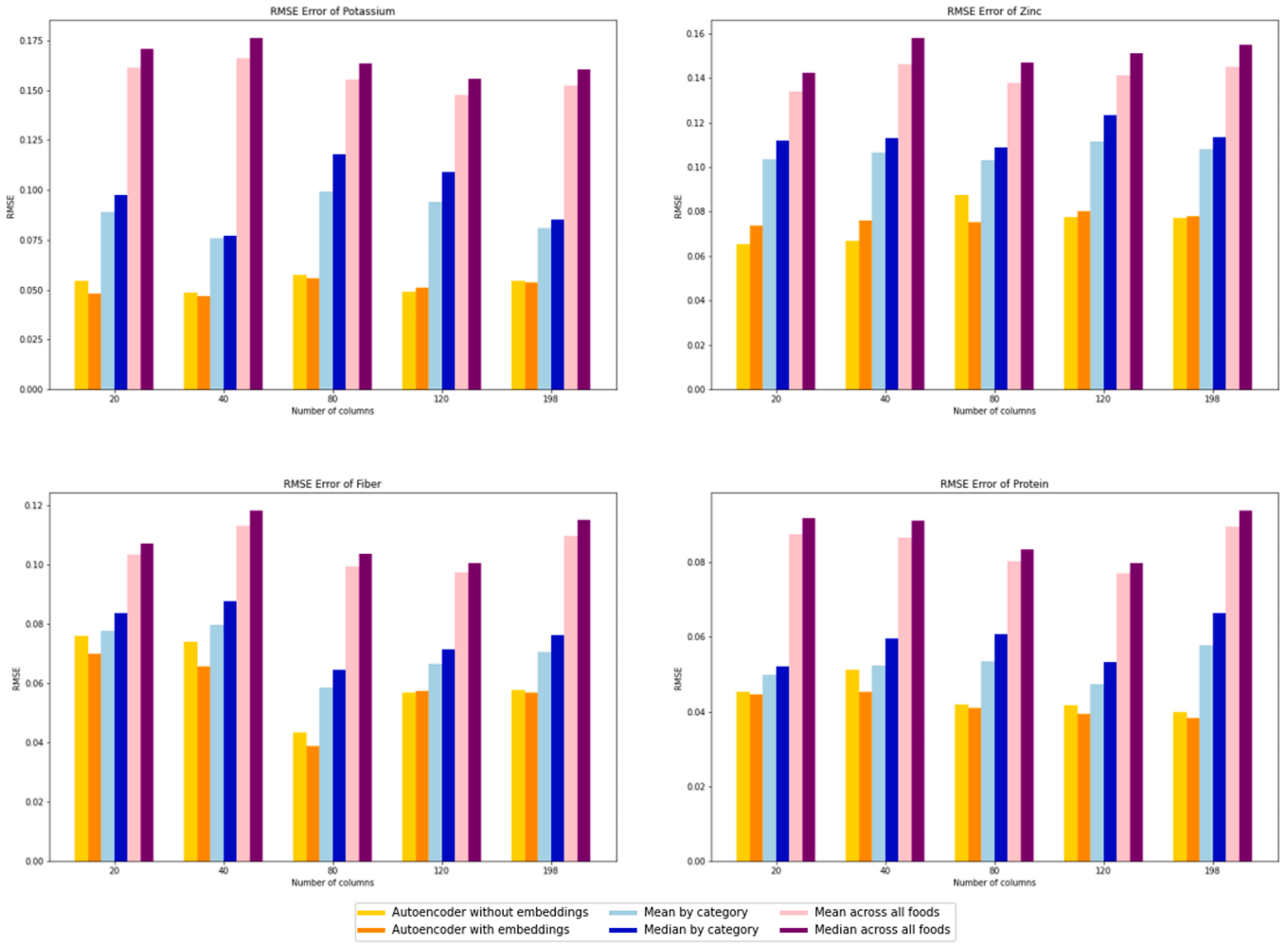


Fig. 3. RMSE results for potassium, zinc, fiber and protein in the average value datasets.

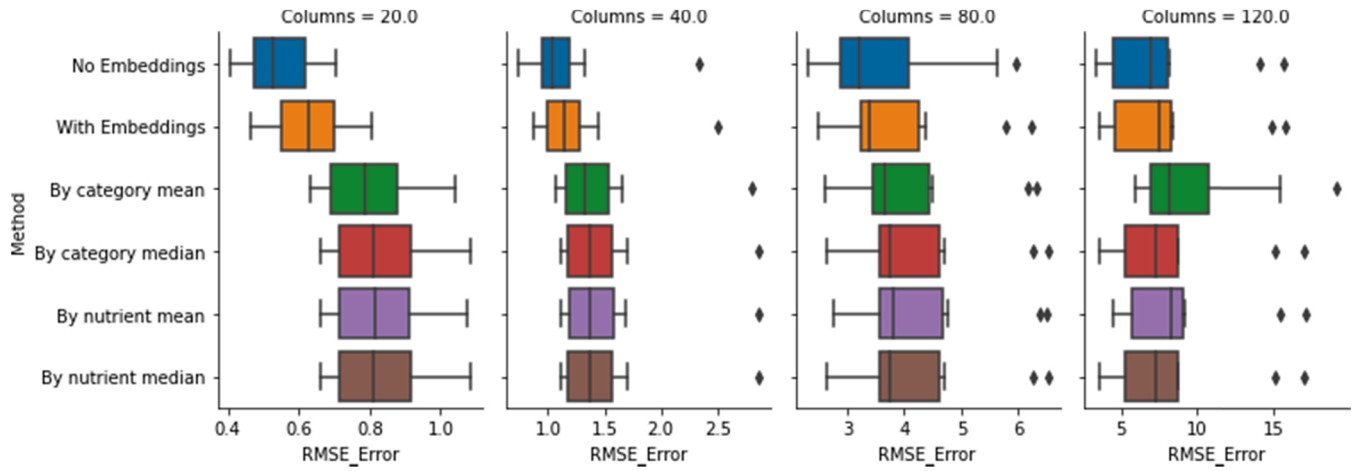
that FCDBs nowadays can contain up to 40% missing data (Greenfield and Southgate, 2003).

In order to explore the robustness of the autoencoders, we repeated the previously explained experiment 10 times, including splitting the data in train and test sets, introduction of missingness in the sets and the training of the autoencoders. All of the methods for missing data imputation were implemented using the Python programming language. The traditional imputation methods based on food category were implemented by making separate dataset for them, the autoencoder was implemented using the 'Pytorch' library (Paszke et al., 2019), and for the imputation of mean and median values across all food items we used the Multiple Imputer from the 'Autoimpute' Python package (Kearney et al., 2021). The output for each of the methods are a new complete dataset with calculated missing values. We then perform the performance measure explained in 3.2 for each method by calculating the difference between the imputed values and the real values in each nutrient of the dataset. The calculations were done on the scaled values to avoid disproportionate attribute contributions.[1]

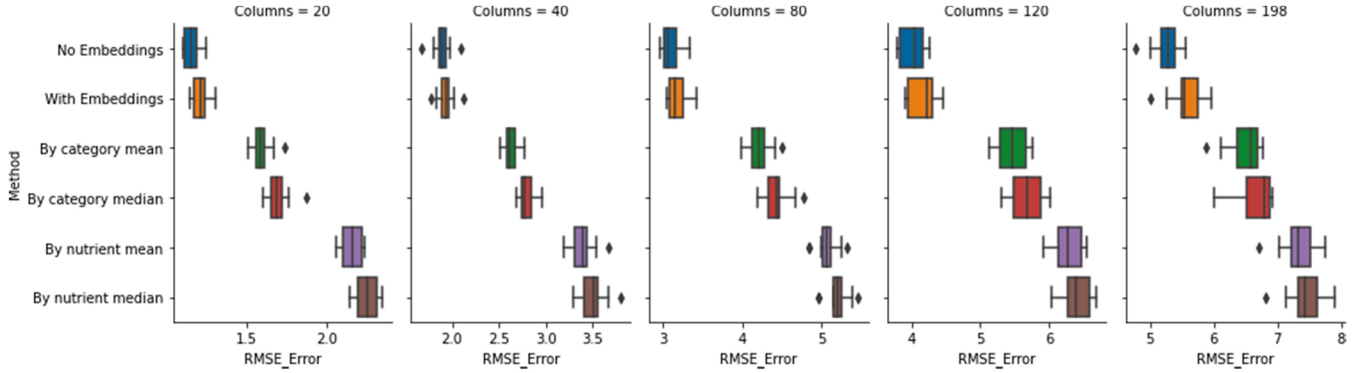
Table 5 shows the average $RMSE_{sum}$ error from 10 iterations for each imputation method for the aforementioned use cases using the median value dataset as a base dataset. From the obtained results, we can deduce that the traditional methods always yield larger sums of errors across all nutrients than the deep learning autoencoder method in every use case. Furthermore, the results also show that adding word embeddings does not improve the overall error produced by the autoencoder. The results of RMSE for certain nutrients shown in Fig. 2 demonstrate similar results as the sum of errors in Table 5. There are several factors that contribute to these results. Firstly, as shown in Table 4, the number of samples in the median datasets for each use case decreases as more nutrients are added. This produces smaller training set with big dimensionality leading to bigger reconstruction error during training of the autoencoder. For the traditional methods, calculating the mean of nutrient values by food categories shows better results than calculating mean across all foods or calculating median. It is important to mention that in some use cases, namely the use case for 120 nutrients, the results show that the median imputation method is better than the mean imputation. This is due to the fact that the train and test datasets are constructed of randomly chosen samples of the median dataset and, as we mentioned in Section 2.1, these datasets are largely populated with zero values which results in bias towards the mean and median imputation methods, and this is especially evident when the original nutrient value is very small i.e. is close to or equal to zero. The variability in reconstruction errors can also be seen in Fig. 4 where the distribution of RMSE error over 10 iterations is presented. The distribution of errors generated over 10 iterations has much higher variability than the errors when testing the methods on the average values datasets.

Table 6 shows the average $RMSE_{sum}$ error for each imputation method for the use cases using the average values dataset as a base dataset and showcase that the results are similar to the results obtained with the median value datasets. In comparison to the median value

[1] The code for the experiments is available at https://github.com/gjorshoskaivana/MIDA-in-FCDBs.



(a) Median dataset



(b) Average dataset

Fig. 4. $RMSE_{sum}$ error for 10 iterations (20% missingness).

Table 6
Average $RMSE_{sum}$ error on average datasets (20% missingness).

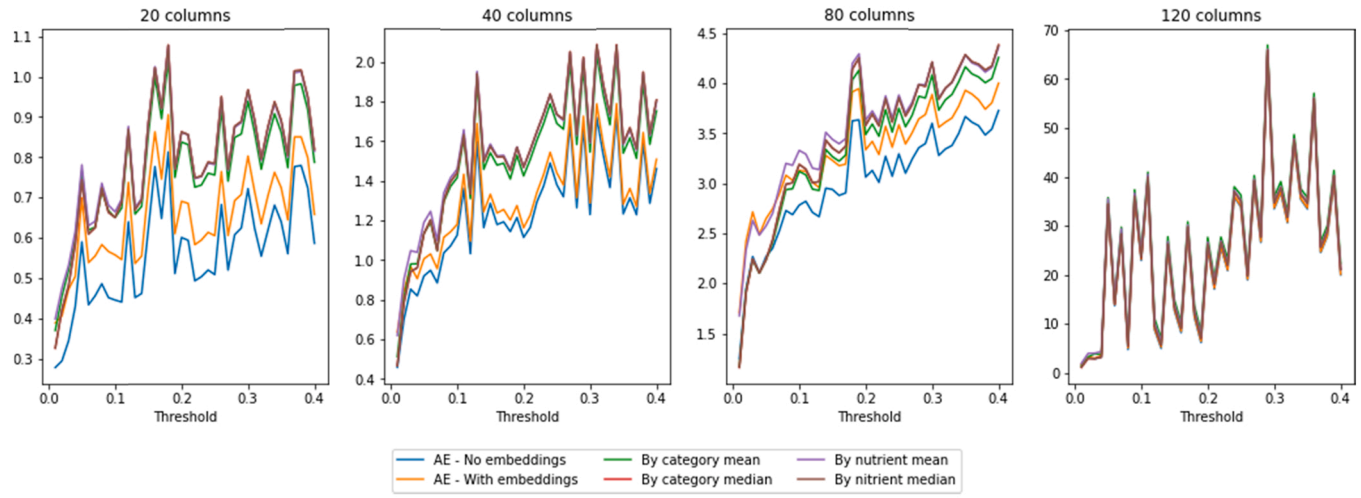
	20 columns	40 columns	80 columns	120 columns	198 columns
Autoencoder without embeddings	**1.15**	**1.89**	**3.09**	**4.01**	**5.26**
Autoencoder with embeddings	1.20	1.93	3.18	4.16	5.57
Mean by category	1.60	2.63	4.23	5.46	6.48
Median by category	1.70	2.79	4.44	5.68	6.65
Mean across all foods	2.16	3.40	5.07	6.27	7.32
Median across all foods	2.26	3.51	5.20	6.40	7.43

dataset, the average value dataset contains significantly more samples without missing values which means that there is more data used for training the autoencoders. Moreover, this dataset contains more non zero values than the median value dataset which is why the results show bigger errors and more consistent results to compare the imputation methods. Fig. 3 shows RMSE for the same nutrients as shown previously in Fig. 2. The graphs show that the autoencoders give better imputation results than the traditional methods. Furthermore, they indicate that adding word embeddings yield better results in certain nutrients, compared to the same results in the median value dataset that almost always showed worse results. We can state that the imputation method with an autoencoder performs better than the mean and median imputations and we can rank the imputation methods according to their $RMSE_{sum}$ values, the most accurate being the imputation with autoencoder without using word embeddings, followed by the autoencoder imputation with word embeddings, imputing mean or median nutrient value by food category, and lastly, imputing mean or median value calculated across all food items.

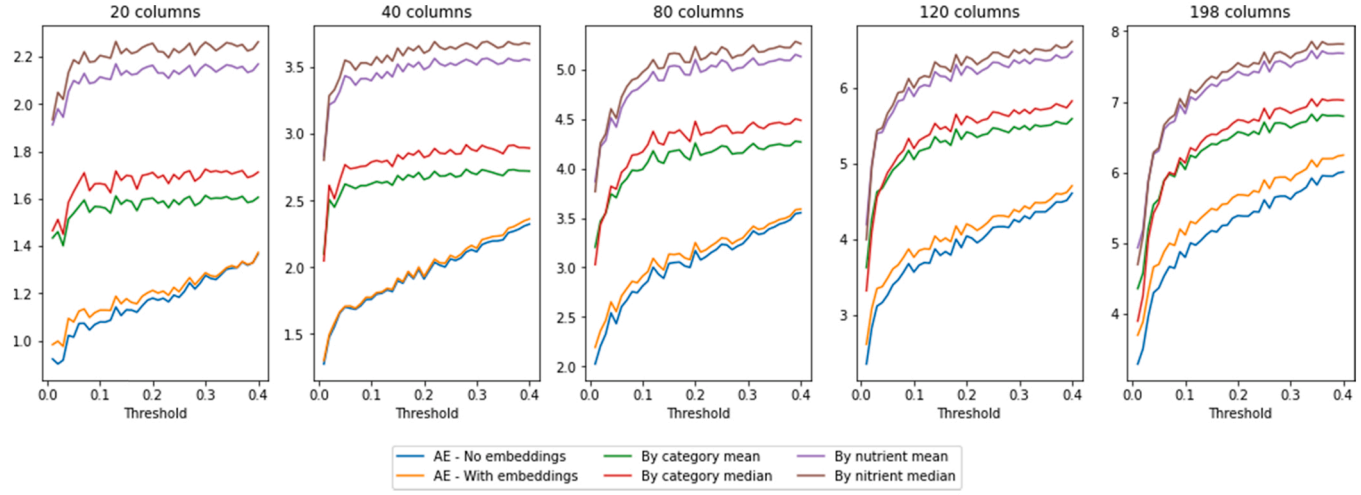
The results from the additional experiment using varying percentages of missingness are shown in Fig. 5. The errors on the average value datasets show consistent rise in correspondence to the increase of the missingness thresholds. Additionally, the average value results show that there is a significantly better performance of the autoencoder models compared to the traditional methods of missing value imputations. On the other hand, the graphs showcasing the performance of the methods on the median value datasets show varying results. However, although there is no consistency in the errors generated for different missingness thresholds, the autoencoders still yield the least amounts of errors across all use cases.

5. Discussion

As mentioned in Section 2.2.1, there are several rules that need to be satisfied in order to ensure that the FCD that is being used to borrow values from is consistent and compatible and due to the difficulty of finding such data, the most common approaches to fill in missing values in FCDBs thus far is using mean or median values from several other FCDBs or computing these values from one FCDB based on the food category of the food item. Both of these approaches are not ideal, namely using only one database to obtain data by food category can lead to very inaccurate values for certain nutrients. As an example, both apples and oranges fall in the same category of ”Fruits”, however on average,



(a) Median dataset



(b) Average dataset

Fig. 5. Average $RMSE_{sum}$ error for 10 iterations on varying missingness thresholds (a) median dataset (b) average dataset.

oranges contain 225 IU of Vitamin A whereas apples contain 54 IU of the same nutrient, meaning that there is a drastic difference in the values and leading to inaccurate value imputations. On the other hand, using several FCDBs to calculate mean and median values is difficult due to their compatibility. Several studies have compared values obtained from chemical analysis with values computed by using FCDBs with varying findings (Stock and Wheeler, 1972; Wolf, 1981; McCullough et al., 1999). In Arab (1985), the difficulty of making international comparison for FCDBs is analyzed and the variations in terminology and composition of foods are identified as the main problems. Because of these issues, it is necessary to find alternative methods of more accurate missing value imputation in FCDBs. One way to do this is to use statistical and classical machine learning methods. Such methods have been explored in (Ispirova et al., 2019, 2020), where it was shown that promising results are achieved. However, they were developed and tested only in scenario where only the value of one nutrient is predicted. In reality, there are a lot of missing nutrient values for the same food item. To go beyond this, the deep learning approach for missing value imputation using Denoising Autoencoders provides more accurate results because these models manage to map nutrient composition vectors in high level representations and learn to predict values based on similarity of the vectors, therefore similar foods are going to have similar values for the respective nutrients. To compare the results of this approach, only the most commonly used approaches have been selected, which are imputations done with classical statistics, either mean or median values. The results have not been compared with the classical machine learning approaches, since they have been developed for scenario where values only from one nutrient is missing, which is not a case in the approach with autoencoders. Comparing them will lead to an unfair comparison, where the experimental design differs.

The biggest challenge while conducting the experiments was the selection of the nutrients. As mentioned in Section 2.1, there are a total of 97,985 food-nutrient pairs whereas in the new datasets there should be 2,594,790 pairs meaning that only 3% of the values needed for constructing the datasets were available. Imputing a zero value in place of the pairs that do not exist in the original dataset taken from the FoodData Central (Agricultural Research Service, 2021) resulted in zero centered distributions of nutrient values. Fig. 6 illustrates the distributions of a few nutrients in the newly constructed datasets for average and median values accordingly and clearly showcase that the nutrient values across all food items are centered at zero. Furthermore, Fig. 7 shows the percentages of non zero values for all nutrients in the median and average datasets and demonstrate that the majority of the nutrients contain between 0% and 2% non zero values across all food items. This creates a significant bias when predicting nutrient values close to zero, since the mean and median values of the nutrients are also either close or equal to zero. Our initial approach was to select a specific number of nutrients from the datasets that contain the least amount of undefined

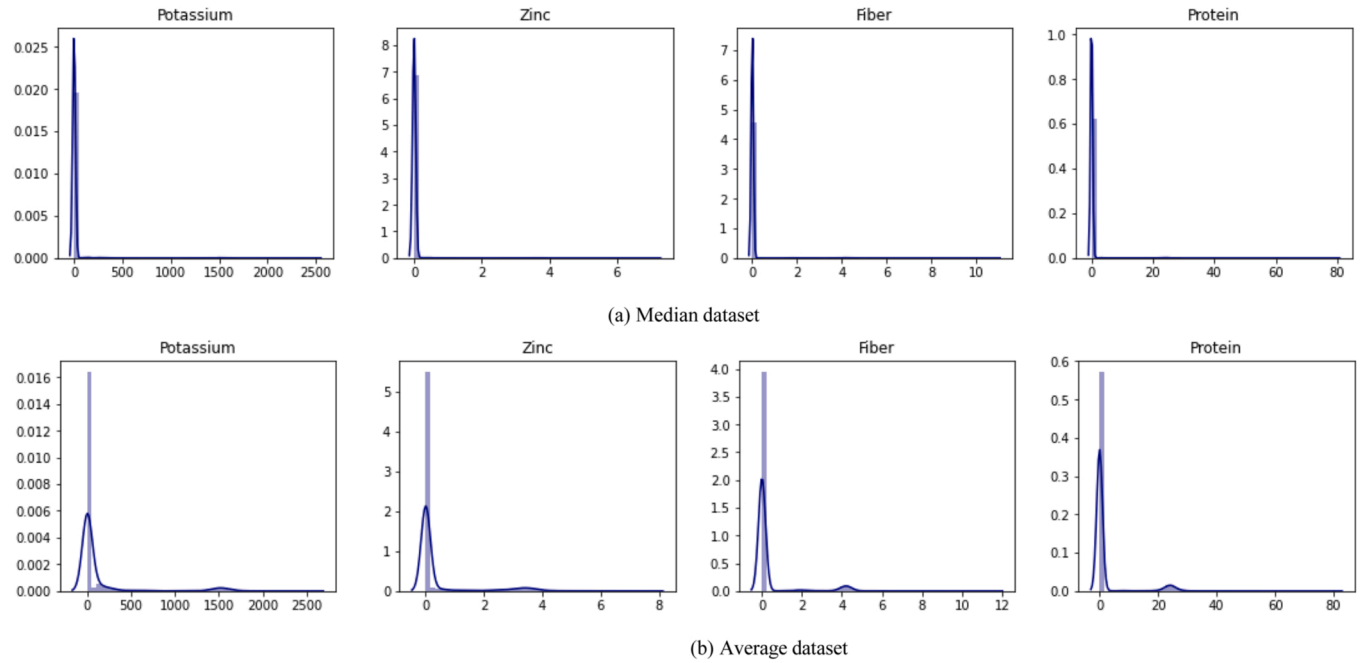


Fig. 6. Distributions of potassium, zinc, fiber and protein values in the datasets (a) median dataset (b) average dataset.

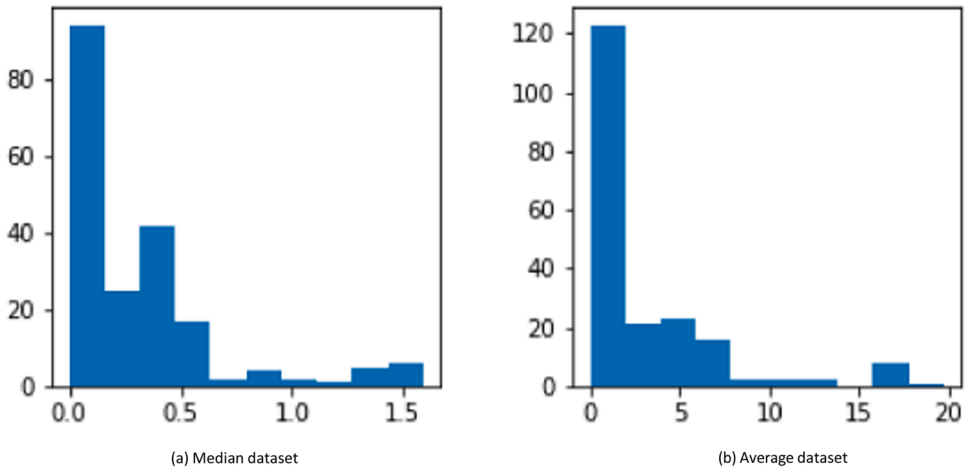


Fig. 7. Histograms of percentages of non-zero values across all nutrients.

values and it proved the bias towards mean and median imputation with $RMSE_{sum}$ reaching zero in certain use cases, with comparable very small errors produced by the autoencoder imputation method. Instead, selecting the nutrients that contain the highest percentage of non zero values produced more variety in numerical values across the food items therefore resulting in better performance of the autoencoder models and accentuated their ability to give more accurate approximation of the missing values.

An interesting fact that was noticed was the improvement in imputation of the autoencoders by adding word embeddings to the food samples in the datasets containing average values. By providing more samples for training that consist of wider variety of values for the nutrients, the autoencoder architecture is capable of learning value approximations more accurately and the word embeddings in this case serve as latent variables. The autoencoder is able to map similar foods with similar values for their respective nutrients more accurately than the traditional imputation methods of filling in mean or median values by food category. Furthermore, the use of word embeddings automates the process of choosing similar foods for missing value imputation that otherwise have to be manually picked to perform mean or median calculation. This advantage of the autoencoders could further be explored and improved by adding more food samples in the training datasets and assigning different, non zero values for the food nutrient value pairs that have not been obtained using food composition analysis.

Taking into account the results we obtained from this study, deep learning architectures such as autoencoders should be taken into strong consideration when dealing with missing data in FCDBs. Among the tested methods, autoencoders showed the best results. Although this approach requires more data for training purposes in order to obtain satisfactory outcomes, it produces a ready-to-use model for predicting missing values in any food sample that contains the suitable nutrients.

6. Conclusion

FCDBs are a powerful information resource used in many domain such as food and nutritional science, food industry for food production and consumption as well as public health. One of their main limitations is the existence of missing data and with the increasing quantity of data,

this becomes an amplifying problem that restrains their usage. The easiest way to deal with missing data in FCDBs is to ignore it, but when the data is necessary, the most used imputation approaches are calculating average and median values from the available data that can come from either the same FCDB or same food from FCDBs of other countries. Despite its ease of application, this method introduces considerable amount of errors particularly due to the fact that there is limited space for errors because of the specific data contained in FCDBs. As a solution, we propose using deep learning algorithms for missing value imputation, specifically autoencoders as an unsupervised deep learning algorithm. We explored whether the state-of the-art methods should be preferred over the traditional methods of imputing calculated average or median values. We examined the imputation results on two datasets with data extracted from USDA FoodData Central FDCBs. The results show that the state-of-the-art method outperform the traditional methods, hence we conclude that Autoencoders can be considered as a method for dealing with missing values in FCDBs.

Author contributions

Ivana Gjorshoska: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Drafting the manuscript, Approval of the version of the manuscript to be published. Tome Eftimov: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Drafting the manuscript, Revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published. Dimitar Trajanov: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Drafting the manuscript, Revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published.

Acknowledgments

This work was supported by the Slovenian Research Agency (research core funding programmes P2-0098) under grant agreement 101005259 (COMFOCUS). This research was undertaken by Jožef Stefan Institute (JSI, SI), a beneficiaryin FNS-Cloud, which has received funding from the European Union's Horizon 2020 Research and Innovation programme (H2020-EU.3.2.2.3. – A sustainable and competitiveagri-food industry) under Grant Agreement No. 863059 – www.fns-cloud.eu.

References

Abiri, N., Linse, B., Edén, P., Ohlsson, M., 2019. Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems. Neurocomputing 365, 137–146.

Agricultural Research Service, U.D.o.A., 2021. FoodData Central. URL: ⟨https://fdc.nal.usda.gov/⟩. (Accessed 11 August 2021).

Arab, L., 1985. Summary of survey of food composition tables and nutrient data banks in europe. Ann. Nutr. Metab. 29, 39–45.

Boquet, G., Vicario, J.L., Morell, A., Serrano, J., 2019. Missing data in traffic estimation: a variational autoencoder imputation method. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 2882–2886.

Camino, R.D., Hammerschmidt, C.A., State, R., 2019. Improving Missing Data Imputation with Deep Generative Models. arXiv preprint arXiv:1902.10666.

Cho, K., 2013a. Boltzmann Machines and Denoising Autoencoders for Image Denoising. arXiv preprint arXiv:1301.3468.

Cho, K., 2013b. Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images. In: Proceedings of the International conference on machine learning, PMLR. pp. 432–440.

Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the Properties of Neural Machine Translation: Encoder-decoder Approaches. arXiv preprint arXiv:1409.1259.

Church, S., 2009. Eurofir synthesis report no 7: food composition explained. Nutr. Bull. 34, 250–272 doi:10.1111/j.1467-3010. 2009.01775.x.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A., 2017. Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Copenhagen, Denmark, pp. 670–680. URL: ⟨https://www.aclweb.org/anthology/D17-1070⟩.

EuroFIR, 2021. List of EuroFIR FCDBs EuroFIR. URL: ⟨https://www.eurofir.org/food-information/food-composition-databases⟩. (Accessed 11 August 2021).

70 Forrest, L., Jukic, K., Gilbert, H., Probst, Y., 2013. Continuing education: advanced food composition data use in practice. Nutr. Diet. 81–83. https://doi.org/10.1111/1747-0080.12027.

Globerson, A., Chechik, G., Pereira, F., Tishby, N., 2007. Euclidean Embedding of Co-occurrence Data.

Gondara, L., 2016. Medical image denoising using convolutional denoising autoencoders. In: Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), IEEE, pp. 241–246.

Gondara, L., Wang, K., 2018. Mida: Multiple imputation using denoising autoencoders. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 260–272.

Greenfield, H., 1995. Quality and accessibility of food-related data. In: Proceedings of the First International Food Data Base Conference, Sydney, Australia, 22–24 September 1993.

Greenfield, H., Southgate, D.A., 2003. Food Composition Data: Production, Management, and Use. Food & Agriculture Organisation.

, 2003Health Canada, H.P., Branch, F., 2018. Canadian Nutrient File Search Engine Online. URL: ⟨https://food-nutrition.canada.ca/cnf-fce⟩. (Accessed 11 August 2021).

Hinton, G.E., Krizhevsky, A., Wang, S.D., 2011. Transforming auto-encoders. In: Proceedings of the International Conference on Artificial Neural Networks, Springer. pp. 44–51.

Ichikawa, M., Hosono, A., Tamai, Y., Watanabe, M., Shibata, K., Tsujimura, S., Oka, K., Hitomi, F., Okamoto, N., Kamiya, M., Kondo, F., Wakabayashi, R., Naoguchi, T., Isomura, T., Imaeda, N., Goto, C., Yamada, T., Suzuki, S., 2019. Handling missing data in an ffq: multiple imputation and nutrient intake estimates. Public Health Nutr. 22, 1–10. https://doi.org/10.1017/S1368980019000168.

Ispirova, G., Eftimov, T., Korošec, P., Korousic Seljak, B., 2019. Might: statistical methodology for missing-data imputation in food composition´ databases. Appl. Sci. 9, 4111.

Ispirova, G., Eftimov, T., Seljak, B.K., 2020. Evaluating missing value imputation methods for food composition databases. Food Chem. Toxicol. 141, 111368.

Kim, J., Chung, K., 2020. Multi-modal stacked denoising autoencoder for handling missing data in healthcare big data. IEEE Access 8, 104933–104943 doi:10.1109/ACCESS.2020.2997255.

Kearney, J., Barkat, S., Bose, A., 2021, Python Package Index – Pypi, Autoimpute. URL: ⟨https://pypi.org/project/autoimpute⟩ [Accessed 20 July 2021].

Lebret, R., Collobert, R., 2013. Word Emdeddings through Hellinger PCA. arXiv preprint arXiv:1312.5542.

Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., Chen, E., 2015. Word embedding revisited: a new representation learning and explicit matrix factorization perspective. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence.

Ma, Q., Lee, W.C., Fu, T.Y., Gu, Y., Yu, G., 2020. Midia: exploring denoising autoencoders for missing data imputation. Data Min. Knowl. Discov. 34, 1859–1897.

Martin, J.H., 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson/Prentice Hall.

McCullough, M.L., Karanja, N.M., Lin, P.H., Obarzanek, E., Phillips, K.M., Laws, R.L., Vollmer, W.M., O'Connor, E.A., Champagne, C.M., Windhauser, M.M., et al., 1999. Comparison of 4 nutrient databases with chemical composition data from the dietary approaches to stop hypertension trial. J. Am. Diet. Assoc. 99, S45–S53.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. Adv. Neural Inf. Process. Syst. 3111–3119.

Miok, K., Nguyen-Doan, D., Robnik-Šikonja, M., Zaharie, D., 2020. Multiple Imputation for Biomedical Data Using Monte Carlo Dropout Autoencoders. arXiv preprint arXiv:2005.06173.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.

Schakel, S., Buzzard, I., Gebhardt, S., 1997. Procedures for estimating nutrient values for food composition databases. J. Food Compos. Anal. 10, 102–114.

Stock, A.L., Wheeler, E.F., 1972. Evaluation of meals cooked by large-scale methods: a comparison of chemical analysis and calculation from food tables. Br. J. Nutr. 27, 439–448.

Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. Adv. Neural Inf. Process. Syst. 3104–3112.

Theis, L., Shi, W., Cunningham, A., Huszár, F., 2017. Lossy Image Compression with Compressive Autoencoders. arXiv preprint arXiv:1703.00395.

, 2017USDA, 2021. USDA. URL: ⟨https://www.usda.gov⟩. (Accessed 26 August 2021).

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A., 2008. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103.

Williamson, C., 2006. The Different Uses of Food Composition Databases: Synthesis Report No. 2. European Food Information Resource Consortium (EuroFIR).
Wolf, W., 1981. Assessment of inorganic nutrient intake from self-selected diets. In: Human Nutrition Research, BARC Symposium No. 4, pp. 175–196.
Zhu, Z., Wang, X., Bai, S., Yao, C., Bai, X., 2016. Deep learning representation using autoencoder for 3d shape retrieval. Neurocomputing 204, 41–50.
Zhuo, L., Cheng, B., Zhang, J., 2014. A comparative study of dimensionality reduction methods for large-scale image retrieval. Neurocomputing 141, 202–210.