Türkiye Klinikleri Biyoistatistik Dergisi
Turkiye Klinikleri Journal of Biostatistics Turkiye Klinikleri J Biostat. 2023;15(1):1-12

ORİJİNAL ARAŞTIRMA ORIGINAL RESEARCH DOI: 10.5336/biostatic.2022-90804

The Effect of Missing Data Mechanisms on Deep Learning in Binary Classification: A Simulation Study

İkili Sınıflandırmada Eksik Gözlem Mekanizmalarının Derin Öğrenmeye Etkisi: Bir Benzetim Çalışması

Ebru ÖZTÜRK[a], Yağmur ZENGİN[a], Merve KAŞIKCI[a], Erdal COŞGUN[b]

[a]Department of Biostatistics, Hacettepe University Faculty of Medicine, Ankara, Türkiye
[b]Genomics Team, Microsoft Research, WA, USA

ABSTRACT Objective: Investigating the effects of missing data and the methods to overcome problems in statistical models caused by missingness is a significant research topic due to the complex nature of the data, which includes missing observations. The different statistical approaches used in the case of the missing data are complete case analysis and missing data imputation. It is necessary to evaluate missing data mechanisms and patterns to handle missing data issues. However, understanding the missing data mechanism is not easy in relatively large data sets. Recently, deep learning algorithms have been widely used for classification, regression, or clustering tasks in large data sets due to computational advances. The objective of this study is to present the effect of missing data mechanisms on the performance of the deep learning algorithm for binary classification problems. **Material and Method:** To achieve the aim of this study, an extensive simulation study was conducted using Virtual Machine on Microsoft Azure by considering the missing proportion, the correlation structure, and the mechanism of the missing in the large data set. For different missing data mechanisms, the performance of deep learning with list-wise deletion and imputation compared to the original data set was investigated. **Results:** It is observed that while the proportion and the mechanism of the missing affect slightly the performance of the deep learning, the correlation level of data affects relatively. **Conclusion:** Although slight differences were obtained from the area under the curve values, deep learning algorithms can overcome the problem caused by missingness in large data sets.

Keywords: Missing data; missing data imputation; missing data mechanism; deep learning

ÖZET Amaç: Eksik gözlemin etkisi ve istatistiksel modellemede eksik gözlem kaynaklı problemlerin çözümü, eksik gözlem içeren verilerin karmaşık yapısı nedeniyle önemli bir araştırma konusudur. Eksik gözlem söz konusu olduğunda kullanılan istatistiksel yöntemler tam gözlemlerin kullanılması ve eksik veri atamasıdır. Eksik veriden kaynaklı problemleri çözebilmek için eksik veri mekanizmalarını ve örüntülerini araştırmak gerekmektedir. Ancak büyük veri kümelerinde eksik veri mekanizmasını ve örüntüsünü anlamak kolay değildir. Son zamanlarda derin öğrenme algoritmaları, teknolojik ilerlemeler nedeniyle büyük veri kümelerinde sınıflandırma, regresyon veya kümeleme görevleri için yaygın olarak kullanılmaktadır. Bu çalışmanın amacı, ikili sınıflandırma problemleri için eksik veri mekanizmalarının derin öğrenme algoritmasının performansı üzerindeki etkisini ortaya koymaktır. **Gereç ve Yöntemler:** Bu çalışmanın amacına ulaşmak için büyük veri setindeki eksik gözlem oranı, korelasyon yapısı ve eksik veri mekanizması dikkate alınarak Microsoft Azure üzerinde Sanal Makine kullanılarak kapsamlı bir simülasyon çalışması yapılmıştır. Farklı kayıp veri mekanizmaları için tam gözlem ve eksik veri ataması yapılan veri kümelerinin orijinal veri kümeleriyle karşılaştırılması yapılmıştır. **Bulgular:** Kayıpların oranı ve mekanizması derin öğrenmenin performansını biraz etkilerken, verilerin korelasyon düzeyinin göreceli olarak etkilediği görülmektedir. **Sonuç:** *Eğri altında* kalan *alan* değerlerinde küçük farklılıklar elde edilmiş olsa da derin öğrenme algoritmaları büyük veri setlerinde eksik veriden kaynaklanan problemin üstesinden gelebilmektedir.

Anahtar kelimeler: Eksik veri; eksik veri atama; eksik veri mekanizması; derin öğrenme

Missing observations in data affect the results due to the decreased data quality and may produce bias depending on the proportion of the missing. There are several sources which can cause the missing, such as database integration issues or system failure in large databases, software-produced errors in research data or

Correspondence: Ebru ÖZTÜRK
Department of Biostatistics, Hacettepe University Faculty of Medicine, Ankara, Türkiye
E-mail: ebru.ztrk3@gmail.com

Peer review under responsibility of Turkiye Klinikleri Journal of Biostatistics.

Received: 22 Apr 2022 ***Received in revised form:*** 30 Sep 2022 ***Accepted:*** 21 Oct 2022 ***Available online:*** 12 Dec 2022

in surveys, sensitive participants who do not answer some questions or drop outs in clinical studies. Handling missing data, especially in epidemiologic or clinical research, while considering the proper methods, which vary due to the underlying missing data structure, is an important task in data science.

The methods which benefit from the recent technological and methodological developments to handle missing data have become more and more popular since complete-case analysis (CCA) may cause the loss of critical information. There are several comparative studies in the literature that investigate the effects of missing estimation techniques when the missing data were imputed. Moreover, stochastic process or fuzzy logic theory and machine learning approaches are very popular on large data sets.

Deep learning could be defined as the hierarchical machine learning technique that is similar in a way to the biological nervous system in terms of information processing and pattern recognition. Deep learning methods consist of multiple levels or layers, i.e., a hierarchical structure. The important characteristic of the deep learning methods is that these layers are automatically constructed by learning from the data. In large databases with a large proportion of missing, deep learning could be a useful tool to increase classification performance compared to the other commonly used approaches.[1]

In this study, we aim to investigate whether the deep learning algorithm can handle missing data without data imputation in a relatively large data set. To accomplish this, we conduct an extensive simulation study to assess the differences between using list-wise deletion and imputation compared to the original simulated dataset in terms of classification performance in a balanced binary classification problem.

MATERIAL AND METHODS

The detailed information about deep learning, missing data and simulation design was demonstrated in this section.

DEEP LEARNING

Deep learning models are extensions of multilayer artificial neural networks (ANNs). The development of advanced inference algorithms increases the computational power while processing large amounts of data and helps to overcome some problems such as overfitting. Hence, the use of complex models such as deep learning methods has increased with successful applications in the last decade. Although the emergence of deep learning dates back to earlier times, its recognition has increased with the publication of articles on its successful implementation.[2-4] The use of deep learning architectures in different fields such as object and speech recognition, natural language processing, text mining, drug design, and bioinformatics has increased.

Deep learning has the ability to learn from a representation, also known as feature learning. Deep learning algorithms carry out the process of learning from new and improved representations of raw data. They complete learning in multiple steps corresponding to their multi-level transformations between hidden layers. Nonlinear subsequent transformations between layers increase the power of expressiveness of the model.[5] Unlike the ANNs, deep learning requires the use of many hidden neurons and strata together with new educational models, which could be defined as an architectural advantage. While the usage of a plurality of neurons permits a comprehensive representation of the available raw data, the layered pipeline of the nonlinear combination of outputs produces a lower dimensional projection of the input area. Each sub-dimensional projection corresponds to a higher level of detection. This characteristic of deep learning results in effective high-level abstraction of raw data or images, provided that the network is optimally weighted.[6] One of the main advantages of deep learning is that it does not require feature selection due to the fact that it has the capacity to carry out feature engineering on its own.

Deep neural network (DNN), which consists of multilayer feed-forward networks, uses the backpropagation algorithm. By using backpropagation, parameters are updated throughout the network.

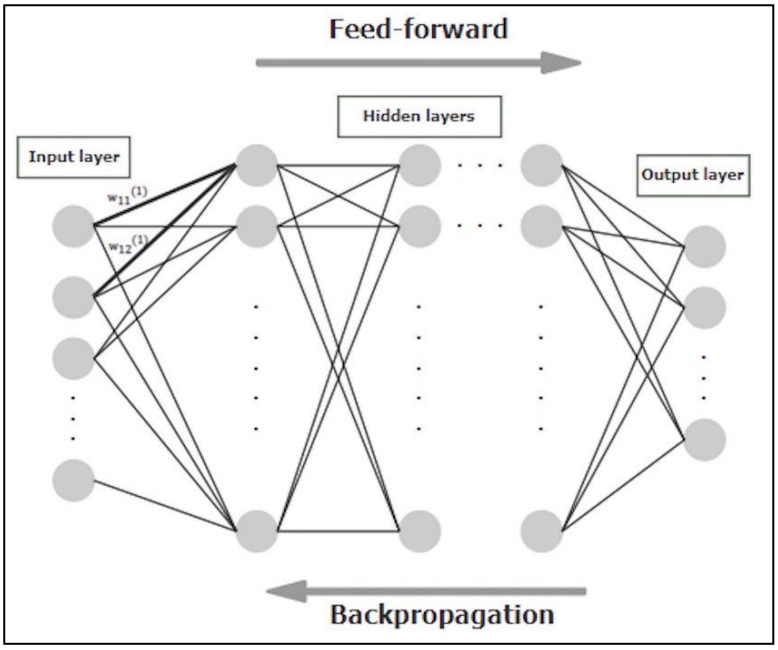

FIGURE 1: Structure of deep neural networks (image source[9]).

The DNN is structured as a deep architecture and consists of sequential layers as in Figure 1. The output of each layer is provided as an input to the next layer. Random values *(w)* are assigned to input neurons as initial weights. The neurons in this structure are linear transformations of the input, obtained by multiplying the input vector (x) by the weight vector and then adding the bias term (b). After the weights of the neurons are calculated, the output is converted to a nonlinear value by using the activation function *(o=f(w×x+b))*. Activation functions are nonlinear, thus allowing the network to approach more complex functions. When the input data has a continuous distribution, it is usually modeled with the rectified linear unit (ReLU) activation function. Optionally, if the network is not too deep, it is recommended to use the hyperbolic tangent (tanh) activation function, especially when ReLU does not provide good results due to the other hyperparameter-related problems in the network.[7]

In the training process of the DNN model, the loss function *(L(w))* used to estimate the difference between model estimation and the actual result is aimed to be minimum. Cross entropy is one of the loss functions that can be used for the classification problem. Let the samples in the training set are represented as *i*, total number of samples as *n*, the actual results as *y*, and the estimated results as ŷ, the cross entropy can be calculated as follows:

$$L(w) = -\sum_{i=1}^{n}[y_i \times \log(\hat{y}_i) + (1 - y_i) \times \log(1 - \hat{y}_i)] \quad (1)$$

To minimize the value obtained from the loss function, the derivative of the function is calculated according to the weight and multiplied by the learning rate (α). Learning rate is a hyperparameter that controls the size of our weight updates, and it is not recommended to use values that are too small or too large. Smaller learning rates require more training time due to small changes in weights with each update, whereas larger learning rates may result in non-optimal weights.[5] The new weight *(w')* is calculated and updated as follows:

$$w' = w - \alpha \times \frac{\partial L(w)}{\partial w} \quad (2)$$

Except learning rate, there are several hyperparameters that can be used to control the weights in DNN model optimization as well. Two of these are *L1* and *L2* regularization parameters that can be added to the

loss function. Among these parameters used as penalty terms, the *L1* is the sum of the absolute values of the weights, whereas the *L2* is the sum of the squares of them.

$$L1:\ \lambda_1 \sum_{j=1}^{p} |w_j| \quad (3)$$

$$L2:\ \lambda_2 \sum_{j=1}^{p} {w_j}^2 \quad (4)$$

Penalty terms have λ_1 and λ_2 constants and they are represented by small values like 10^{-4}, 10^{-5}.[8] Adding *L1* to the loss function reduces the weights by a constant amount toward zero while adding *L2* the weights decrease by an amount proportional to weights. Thus, they provide that the DNN has relatively small weights. Therefore, *L1* and *L2* parameters penalize the large weights. However, when a weight takes on relatively large values, the *L1* shrinks the weight much less than the *L2*. When the weight takes small values, *L1* reduces the weight much more than *L2*.[9] Unlike *L2*, *L1* drives some weights to zero, i.e., it can be used as a feature selection method. Therefore, using *L1* regularization is often preferred because it builds sparse models. For the optimization of the deep learning model, it is important to choose appropriate values for hyperparameters such as number of neurons, number of hidden layers, learning rate, *L1* and *L2*.

Other important hyperparameters to be considered are epoch and batch. Epoch is that all training data passes through the network once. More than one passing of data through the network will increase the generalizability of the model. Including all the data in the analysis at the same time during the training of the model can be challenging in terms of time, speed, and memory. This problem can be overcomed by dividing the data set into parts called batches. For each batch, all the steps in the algorithm are applied. The iteration number indicates that the application of the algorithm is completed for how many batches in these steps.

AN OVERVIEW OF MISSING DATA PATTERN AND MISSING DATA MECHANISM

The missing data is an inevitable issue in social, behavioral, and medical sciences.[10] Little and Rubin defined missing data as the unobserved values of samples.[11] Besides, they claim that those unobserved values carry meaningful importance on the statistical analysis.

Missing data patterns and missing data mechanisms are two different subjects that researchers might confuse. Missing data patterns concern the location of the missing data and do not clarify the reason for missing. Univariate, unit nonresponse, monotone, general, planned missing, and latent missing patterns are the major missing data patterns. In Figure 2, where the black shaded area represents missing observations, these patterns are visualized with any five variables in data matrix written as X_1, X_2, X_3, X_4, and X_5, respectively.[10] The univariate pattern occurs when the missing values are restricted to only one variable. Moreover, the unit nonresponse pattern is observed mostly in survey data and includes the missing data in more than one variable in the same observations. While the monotone pattern contains an increasing trend of missing data for the same observations that happen in longitudinal data mostly, the general pattern, which is observed most commonly, has dispersed missing data. Although the general pattern seems random, it might have systematic missing due to the relationship between variables. The planned pattern has intentional missing values on the data set. Lastly, the latent missing pattern indicates that the entire observations in latent variables are missing and usually confronts the structural equation models.[10]

Missing data mechanisms characterize the relationship between observed and missing values of the variables.[11] The missing mechanisms are missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR).

$Z = z_{ij}$, $M = m_{ij}$ and ϕ are complete data matrix, which includes both dependent and independent variables, indicator matrix of missingness, and unknown parameters, respectively. The missingness mechanism might be written as $f_{M|Z}$ (m_i z_i, ϕ). MCAR might be defined as missingness is related to neither observed nor missing values of the data.[11] MCAR mechanism is:

$$f_{M|Z}(m_i|z_i, \phi) = f_{M|Z}(m_i|\phi) \quad (5)$$

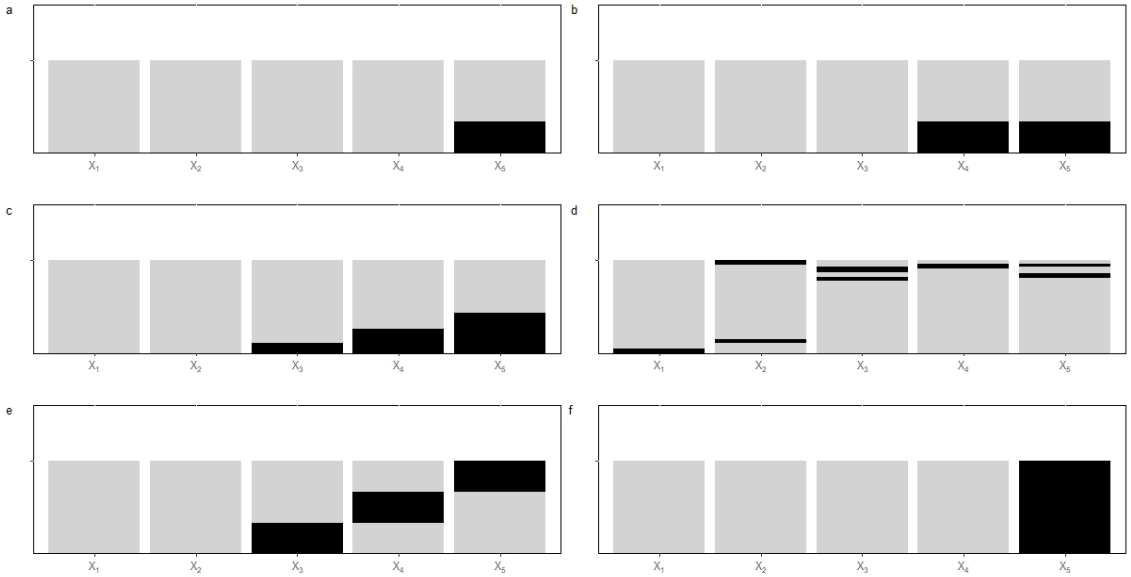

FIGURE 2: Missing data pattern. (a) Univariate pattern. (b) Unit nonresponse pattern. (c) Monotone pattern. (d) General pattern. (e) Planned missing pattern. (f) Latent missing pattern. (image source[10]).

MAR assumes that missingness depends on the observed data. $z_{(0)i}$ and $z_{(1)i}$ are observed and missing parts of the data. MAR mechanism is:

$$f_{M|Z}(m_i|z_i,\phi) = f_{M|Z}\left(m_i\middle|z_{(0)i},\phi\right) \quad (6)$$

MNAR supposes that missingness relies on the missing data. MNAR mechanism is:

$$f_{M|Z}(m_i|z_i,\phi) = f_{M|Z}\left(m_i\middle|z_{(1)i},\phi\right) \quad (7)$$

While one of the common approaches for dealing with missing data issues is CCA, which is also known as list-wise deletion, excludes observations with missing, the other one is imputation, which fills in the missing. There exist different imputation approaches in literature. The milestones of imputation methodology are maximum likelihood estimation and multiple imputation.[12,13]

In this study, the imputation method based on the expectation-maximization with a bootstrap (EMB) algorithm by the Amelia II package is used.[14] Compared to Markov Chain Monte Carlo approaches, the EMB algorithm clearly performs with a relatively large number of variables and a relatively large number of observations in less amount of time. Besides, it offers fundamentally the same results. The main assumption for EMB is that both observed and unobserved data are distributed as multivariate normal.[14] The steps of algorithm of EMB (Figure 3):

(i) The bootstrap samples are drawn from the incomplete data. These samples include missing data to consider the uncertainty in estimation.

(ii) The EM algorithm is performed for each bootstrap sample. After this process, the imputed data set is obtained.

(iii) The separate results are achieved.

(iv) The separate results are combined.

We performed the simulation study for the binary classification problem in large data sets with deep learning and its optimization. Although the multiple imputation approach has robust results, we preferred the EMB algorithm with single imputation to avoid extensive computational load.

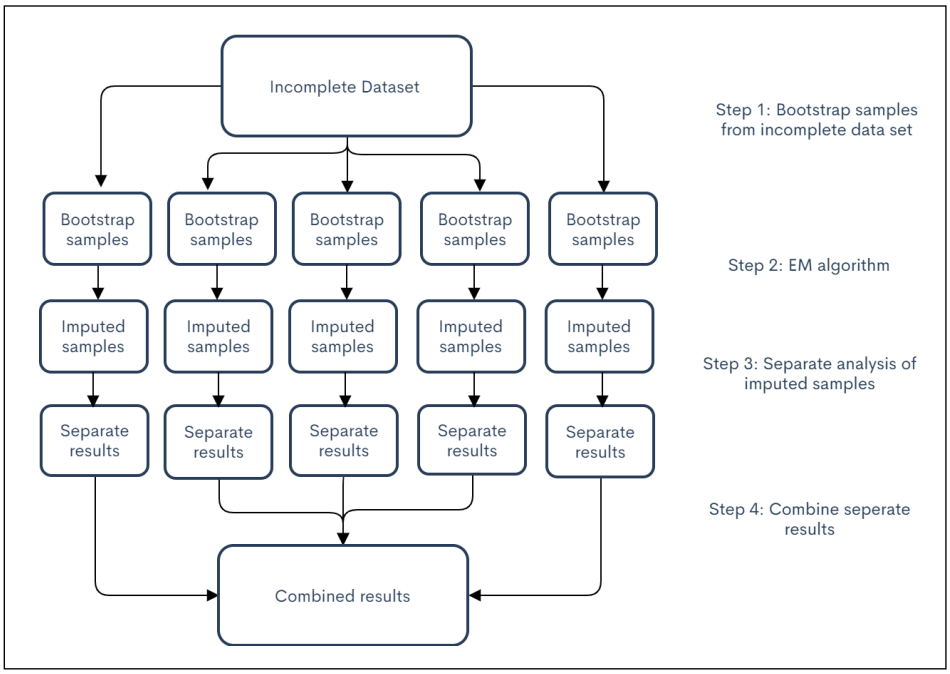

FIGURE 3: The schema for EMB algorithm (image source[14]).
EMB: Expectation-maximization with a bootstrap.

SIMULATION STUDY

In this study, we generated a cross-sectional data set from a multivariate normal distribution with 500 independent variables with binary response, simultaneously considering marginal characteristics and association structure of binary and normal variables with different levels of correlation structure by the BinNor package due to the formerly indicated main assumption of the EMB algorithm.[15,16] The sample size of the data set and the class distribution of response variable are specified as 10,000 and balanced, respectively. To show and validate the model performance, we use the holdout technique, which is the data set into train and test data sets with 80% and 20% of the data set, respectively. To optimize parameters, grid search with 5-fold cross-validation on train data set was applied.

From this generated cross-sectional data set, named as original data, we create missing data with different missing mechanisms (amputated data) with a general missing data pattern by using mice package on both response and independent variables.[17] By using the EMB algorithm in the Amelia II package, we impute each amputated data set.[14] The deep learning algorithm was performed for the original, amputated, and imputed data sets in h2o package in raw data that is there are no preprocessing steps.[18] All possible combinations of the below items represent simulation scenarios:

(i)Data set: Original data set, data set with missing, and imputed data set,

(ii) Missing data mechanisms: MCAR, MAR, and MNAR,

(iii) Missing proportion: 10% and 40%,

(iv) The level of correlation structure in data matrix: Low (ρ=0.3), medium (ρ=0.5) and high (ρ=0.7).

For parameter optimization, we implement grid search by the h2o.grid function, which was used to determine the appropriate number of neurons (100 or 200) and layers (2 or 7). As an activation function tanh, was selected since it permits faster convergence.[8] *L1* and epoch parameters were not optimized due to time limitation. Considering the references, *L1* regularization parameter and the number of epochs were set to 10^{-5} and 10, respectively.[8,19,20]

TABLE 1: Configuration of the virtual machine.

Name	vCPU	Memory: GiB	Temp storage (SSD) GiB	GPU	GPU memory: GiB	Max data disks	Max uncached disk throughput: IOPS/MBps	Max NICs
Standard_NC24s_v3	24	448	2948	4	64	32	80000/800	8

CPU: Central processing unit; Solid state drive; GPU: Graphics processing units; IOPS: Input/output operations per second; NICs: Network interfaces.

In each simulation scenario, the number of repetitions was 1,000. We compared the results of simulations considering the area under the curve (AUC) of the receiver operating curve of the probabilities in the classification performance.

In this study, since extensive computational sources are required, we performed the simulations on the cloud environment. Modern *graphics processing units* are always equipped with a flexible memory hierarchy consisting of various types of memory circuits with different disk latencies and read/write performance.[21] We used cloud technology because of its reliability and accuracy. We've used Microsoft Azure Cloud Computing (Microsoft, USA) for the simulation studies. Configuration of the Virtual Machine (VM) that we used shown on Table 1. Our implementation started with deployment of 4 VMs and installation of RStudio (RStudio Team, Boston, MA). Storage disks configured with automatic scaling. Disk throughput is measured in input/output operations per second and MBps where MBps=10^6 bytes/sec. VMs are also equipped with Intel Xeon E5-2690 v4 (Broadwell) (Intel Corporation, USA) central processing units.[22] The mean and standard deviation for the computational time of the one repetition was calculated at 55.4 and 7.28 minutes.

To ensure the repeatability, the codes of the simulation study and detailed descriptive statistics obtained from repetitions are available at https://github.com/ebozturk/mddl/tree/main/Rnotebooks.

RESULTS

The performance of simulation scenarios was represented with the value of the AUC with boxplots considering for train and test data sets by using the ggplot2 package in Figure 4 and Figure 5 in which they are represented with the missing proportion in rows and correlation structures in columns.[23] The detailed descriptive statistics of the results of simulation scenarios might be found in Appendix 1 and Appendix 2.

When the simulation results were examined in detail, the first notable finding is that the value of the AUC was lower in the test data sets than the train data sets in case of low and medium level correlation in data, while this difference did not appear with high correlation level. It indicates that there might be overfitting problem when the correlation level is between low-to-medium. Moreover, this issue might be related to the curse of dimensionality. The missing proportion of the data set had no or little effect on the AUC in both train and test data sets. The most remarkable results of this study are that the performances of the original data set, the data set with created missing values from this original data set, and the imputed data set from this data set with missing values were close to each other in terms of AUC. Yet, the missing mechanism did not change performances both in the imputed data set and data set with missing. Therefore, it is not possible to state that one of the data set performance is better than the others significantly for different levels of correlation structure, missing data mechanism and missing data proportion in data set.

Overall, one might conclude that deep learning algorithms might overcome the missingness in binary classification problem in terms of performance of the classification in the large data sets.

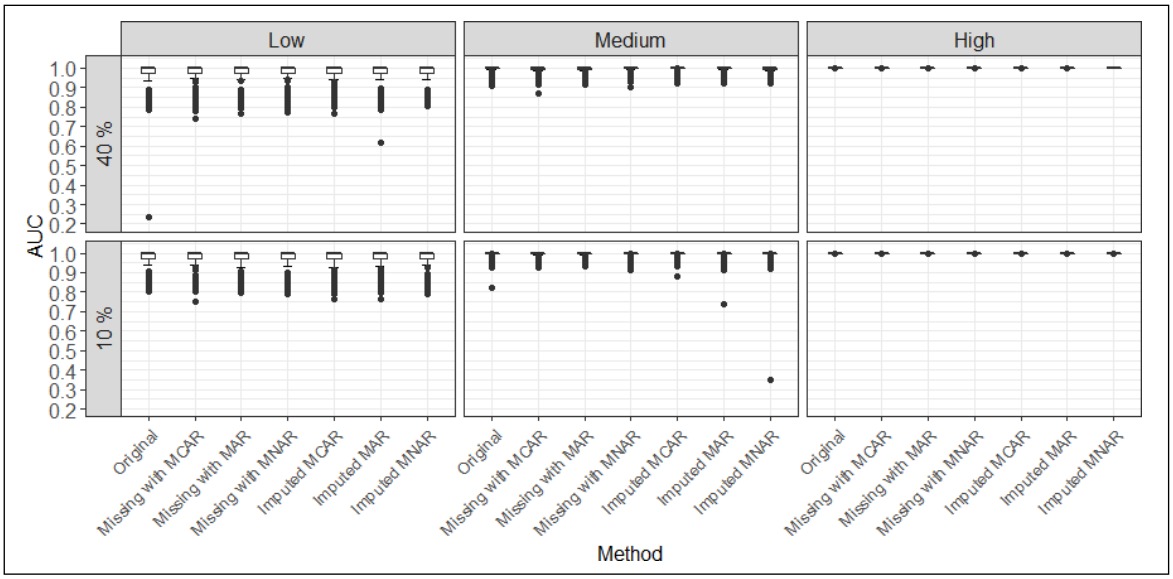

FIGURE 4: Simulation results of train data set.
AUC: Area under the curve; MCAR: Missing completely at random; MAR: Missing at random; MNAR: Missing not at random.

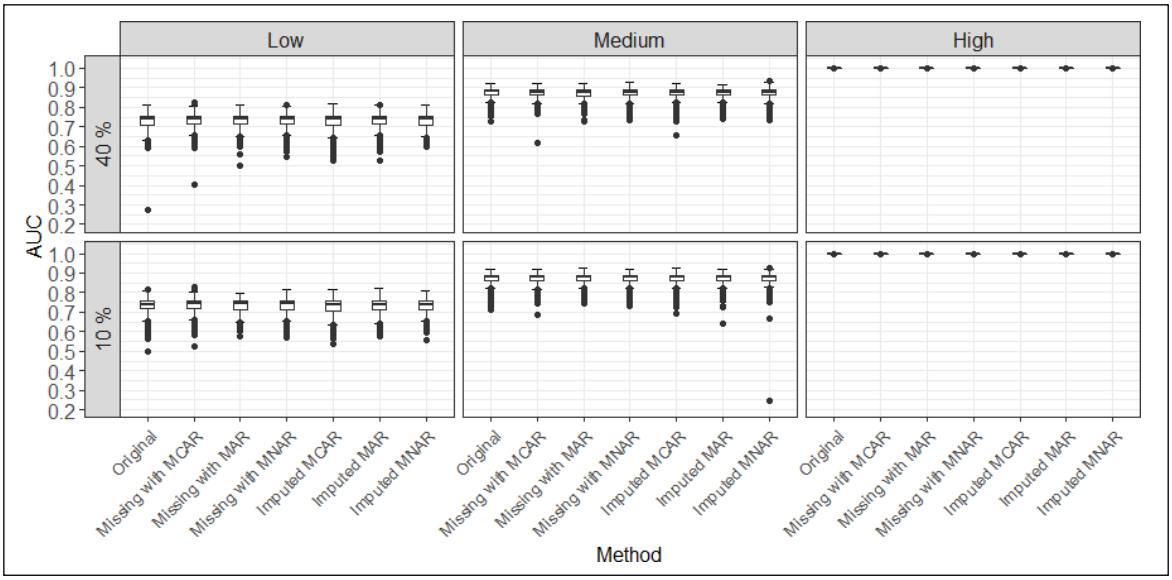

FIGURE 5: Simulation results of test data set.
AUC: Area under the curve; MCAR: Missing completely at random; MAR: Missing at random; MNAR: Missing not at random.

DISCUSSION

Although there are many studies using deep learning to deal with missing observations in the literature, most of these studies aim to impute the missing observations using deep learning structures. However, without imputation, deep learning algorithms can perform well in terms of classification without the complexity of preprocessing. In our study, we aim to investigate how deep learning would provide classification results without imputing missing observations. For this purpose, the classification performance of deep learning was compared using the original data set, imputed data set and data set with missing observations.

In large data sets, it can be difficult to know the missing observation mechanism. In this study, we desire to see the effects of the missing observation mechanisms being MCAR, MAR, or MNAR on classification performance. In particular, when the missing mechanism is MNAR, we aim to see the performance of the deep learning in a larger data set compared to the MCAR and MAR mechanisms.

As a result, it has been seen that the classification performance of deep learning had similar results in different missing data mechanisms in relatively large data set. In addition, the results of the original data set, imputed data set and data set with missing observations were close. Therefore, it can be said that deep learning can also be used in data that is not imputed, and imputation may not be necessary. The same results are valid for different correlation structures and missing proportions investigated in this study. Similarly, Köse et al. used incomplete data set and different imputation methods using a similar approach on the real data set, and the performance of the incomplete data set was not very low compared to the others.[24]

Also, we observed an overfitting problem in the results for low correlation structures. Due to time limitation, we couldn't optimize some parameters, such as L1. In terms of regularization, correlation structure is important along with the optimization of the regularization parameters.[25]

Naqvi et al. conducted a simulation study on the classification of clinical mastitis in cows with and without missing values in the input variables.[26] They concluded that deep learning performance is still good in the scenarios with missing values.

Kia et al. proposed a neural network-based approach that could be used for classification and regression without imputation.[27]

Ghorbani and Zou examined how deep learning would perform in incomplete data sets by including different missing data mechanisms and missing proportions using synthetic data set without simulation.[28] They embedded an extra layer in DNN that represents the pattern of missingness. Their study also indicated that deep learning can handle incomplete data set.

CONCLUSION

To conclude, the objective of this study is to show the effect of the missing data mechanisms and missing data imputation on the performance of the deep learning algorithm in a binary classification problem when the data is relatively large. To reach this aim, an extensive simulation study was conducted. The first step of the simulation study was that we generated a complete data set with respect to different correlation structures. In the second step, we created missing data from this original data set based on MCAR, MAR, and MNAR mechanisms and different missing proportions. After creating missing data, we imputed those missing values by using the EMB algorithm. In the final step, we performed the DNN algorithm to the original data set, data set with missing values and imputed data set to compare the performances of deep learning in terms of classification. Although CCA ignores significant information from incomplete data, the performances of both the imputed data set and data set with missing values are close to the original data set in our study.[29] Therefore, we recommend that deep learning algorithm might handle missing data with large sample sizes without imputation.

In future studies, the same simulation scenarios might be implemented with smaller data set. Due to the extensive computation, we limited our study to a balanced classification problem and general missing pattern. Hence, the same scenarios might be implemented for imbalanced data and different missing data patterns. To prevent overfitting, the optimization of parameters and comparisons of regularization methods might be investigated differently with respect to different correlation structures.

Source of Finance

During this study, no financial or spiritual support was received neither from any pharmaceutical company that has a direct connection with the research subject, nor from a company that provides or produces medical instruments and materials which may negatively affect the evaluation process of this study.

Conflict of Interest

No conflicts of interest between the authors and/or family members of the scientific and medical committee members or members of the potential conflicts of interest, counseling, expertise, working conditions, share holding and similar situations in any firm.

Authorship Contributions

Idea/Concept: *Ebru Öztürk, Yağmur Zengin, Merve Kaşıkcı, Erdal Coşgun;* ***Design:*** *Ebru Öztürk, Yağmur Zengin, Merve Kaşıkcı;* ***Control/Supervision:*** *Ebru Öztürk, Erdal Coşgun;* ***Data Collection and/or Processing:*** *Ebru Öztürk, Erdal Coşgun;* ***Analysis and/or Interpretation:*** *Ebru Öztürk, Yağmur Zengin, Merve Kaşıkcı;* ***Literature Review:*** *Ebru Öztürk, Yağmur Zengin, Merve Kaşıkcı;* ***Writing the Article:*** *Ebru Öztürk, Merve Kaşıkcı, Yağmur Zengin.*

REFERENCES

1. Leke CA, Marwala T. Deep Learning and Missing Data in Engineering Systems. 1st ed. Switzerland: Springer Cham; 2019. [Crossref]
2. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Comput. 2006;18(7):1527-54. [Crossref] [PubMed]
3. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504-7. [Crossref] [PubMed]
4. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012;25:1097-105. [Link]
5. Emmert-Streib F, Yang Z, Feng H, Tripathi S, Dehmer M. An introductory review of deep learning for prediction models with big data. Front Artif Intell. 2020;3:4. [Crossref] [PubMed] [PMC]
6. Ravi D, Wong C, Deligianni F, Berthelot M, Andreu-Perez J, Lo B, et al. Deep learning for health informatics. IEEE J Biomed Health Inform. 2017;21(1):4-21. [Crossref] [PubMed]
7. Patterson J, Gibson A. Deep Learning: A Practitioner's Approach. 1st ed. Sebastopol, CA: O'Reilly Media, Inc; 2017.
8. Candel A, Parmar V, LeDell E, Arora A. Deep Learning with H2O. 6th ed. USA: H2O. ai Inc; 2016. [Link]
9. Nielsen MA. Neural Networks and Deep Learning. San Francisco, CA: Determination Press; 2015.
10. Enders CK. Applied Missing Data Analysis. 1st ed. New York: Guilford Press; 2010.
11. Little RJ, Rubin DB. Statistical Analysis with Missing Data. 3rd ed. New Jersey: John Wiley & Sons; 2019. [Crossref]
12. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological). 1977;39(1):1-38. [Crossref]
13. Rubin DB. Multiple imputations in sample surveys. Proceedings of the Survey Research Methods Section of the American Statistical Association. 1978. [Link]
14. Honaker J, King G, Blackwell M. Amelia ii: A program for missing data. Journal of Statistical Software. 2011;45(7):1-47. [Crossref]
15. Demirtas H, Doganay B. Simultaneous generation of binary and normal data with specified marginal and association structures. J Biopharm Stat. 2012;22(2):223-36. [Crossref] [PubMed]
16. Demirtas H, Amatya A, Doganay B. Binnor: An r package for concurrent generation of binary and normal data. Communications in Statistics-Simulation and Computation. 2014;43(3):569-79. [Crossref]
17. van Buuren S, Groothuis-Oudshoorn K. mice: Multivariate imputation by chained equations in r. Journal of Statistical Software. 2011;45(3):1-67. [Crossref]
18. LeDell E, Gill N, Aiello S, Fu A, Candel A, Click C, et al. h2o: R Interface for the 'H2O' Scalable Machine Learning Platform. R package version 3.30.0.7. 2020. [Link]
19. Pricope TV. A contextual analysis of multi-layer perceptron models in classifying hand-written digits and letters: limited resources. arXiv preprint. 2021. [Link]
20. Singh R, Srivastava S. Stock prediction using deep learning. Multimedia Tools and Applications. 2017;76(18):18569-84. [Crossref]
21. Zhu Y, Wang B, Deng Y. Massively parallel logic simulation with gpus. ACM Transactions on Design Automation of Electronic Systems (TODAES). 2011;16(3):1-20. [Crossref]
22. Microsoft [Internet]. © Microsoft 2022. Ncv3-series. [Cited: October, 2020] Available from: [Link]
23. Wickham H. ggplot2: Elegant Graphics for Data Analysis. 2nd ed. Switzerland: Springer-Verlag New York; 2016.
24. Köse T, Özgür S, Coşgun E, Keskinoğlu A, Keskinoğlu P. Effect of missing data imputation on deep learning prediction performance for vesicoureteral reflux and recurrent urinary tract infection clinical study. Biomed Res Int. 2020;2020:1895076. [Crossref] [PubMed] [PMC]

25. Yu K, Xie W, Wang L, Li W. ILRC: a hybrid biomarker discovery algorithm based on improved L1 regularization and clustering in microarray data. BMC Bioinformatics. 2021;22(1):514. [Crossref] [PubMed] [PMC]
26. Naqvi SA, King MT, DeVries TJ, Barkema HW, Deardon R. Data considerations for developing deep learning models for dairy applications: A simulation study on mastitis detection. Computers and Electronics in Agriculture. 2022;196:106895. [Crossref]
27. Kia SM, Rad NM, van Opstal D, van Schie B, Marquand AF, Pluim J, et al. PROMISSING: Pruning missing values in neural networks. arXiv preprint. 2022. [Link]
28. Ghorbani A, Zou JY. Embedding for informative missingness: Deep learning with incomplete data. 2018 56th Annual Allerton Conference on Communication: Control, and Computing (Allerton). IEEE. 2018:437-45. [Crossref]
29. Steyerberg EW. Clinical Prediction Models: A Practical Approach to Development, Validation and Updating. 2nd ed. Switzerland: Springer International Publishing; 2019.

APPENDIX 1: Descriptive statistics of simulation results in train data set.

Missing proportion		10%			40%		
Correlation structure	**Method**	**$\bar{X}$ (S)**	**M (Q1-Q3)**	**Minimum-maximum**	**$\bar{X}$ (S)**	**M (Q1-Q3)**	**Minimum-maximum**
Low	Original	0.975 (0.048)	1 (0.969-1)	(0.8-1)	0.974 (0.053)	1 (0.972-1)	(0.236-1)
	Missing with MCAR	0.977 (0.046)	1 (0.972-1)	(0.753-1)	0.979 (0.045)	1 (0.976-1)	(0.743-1)
	Missing with MAR	0.974 (0.049)	1 (0.97-1)	(0.799-1)	0.98 (0.043)	1 (0.975-1)	(0.763-1)
	Missing with MNAR	0.973 (0.05)	1 (0.969-1)	(0.789-1)	0.979 (0.045)	1 (0.977-1)	(0.774-1)
	Imputed MCAR	0.975 (0.047)	1 (0.969-1)	(0.761-1)	0.978 (0.045)	1 (0.974-1)	(0.767-1)
	Imputed MAR	0.975 (0.048)	1 (0.971-1)	(0.761-1)	0.979 (0.045)	1 (0.974-1)	(0.62-1)
	Imputed MNAR	0.976 (0.047)	1 (0.973-1)	(0.789-1)	0.98 (0.042)	1 (0.975-1)	(0.806-1)
Medium	Original	0.996 (0.011)	1 (0.999-1)	(0.822-1)	0.996 (0.01)	1 (0.997-1)	(0.909-1)
	Missing with MCAR	0.996 (0.009)	1 (0.995-1)	(0.928-1)	0.996 (0.012)	1 (0.996-1)	(0.868-1)
	Missing with MAR	0.996 (0.01)	1 (0.995-1)	(0.933-1)	0.996 (0.01)	1 (0.996-1)	(0.915-1)
	Missing with MNAR	0.996 (0.01)	1 (1-1)	(0.912-1)	0.997 (0.009)	1 (0.997-1)	(0.899-1)
	Imputed MCAR	0.996 (0.01)	1 (0.999-1)	(0.883-1)	0.997 (0.009)	1 (1-1)	(0.925-1)
	Imputed MAR	0.996 (0.013)	1 (0.999-1)	(0.735-1)	0.997 (0.008)	1 (0.997-1)	(0.924-1)
	Imputed MNAR	0.996 (0.023)	1 (0.999-1)	(0.351-1)	0.996 (0.01)	1 (0.995-1)	(0.924-1)
High	Original	1 (<0.001)	1 (1-1)	(1-1)	1 (<0.001)	1 (1-1)	(1-1)
	Missing with MCAR	1 (<0.001)	1 (1-1)	(1-1)	1 (<0.001)	1 (1-1)	(1-1)
	Missing with MAR	1 (<0.001)	1 (1-1)	(1-1)	1 (<0.001)	1 (1-1)	(1-1)
	Missing with MNAR	1 (<0.001)	1 (1-1)	(1-1)	1 (<0.001)	1 (1-1)	(1-1)
	Imputed MCAR	1 (<0.001)	1 (1-1)	(1-1)	1 (<0.001)	1 (1-1)	(0.999-1)
	Imputed MAR	1 (<0.001)	1 (1-1)	(1-1)	1 (<0.001)	1 (1-1)	(0.999-1)
	Imputed MNAR	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (1-1)	(0.999-1)

MCAR: Missing completely at random; MAR: Missing at random; MNAR: Missing not at random.

APPENDIX 2: Descriptive statistics of simulation results in test data set.

Missing proportion		10%			40%		
Correlation structure	**Method**	**$\bar{X}$ (S)**	**M (Q1-Q3)**	**Minimum-maximum**	**$\bar{X}$ (S)**	**M (Q1-Q3)**	**Minimum-maximum**
Low	Original	0.73 (0.04)	0.741 (0.716-0.755)	(0.501-0.817)	0.726 (0.043)	0.738 (0.705-0.755)	(0.277-0.81)
	Missing with MCAR	0.73 (0.04)	0.742 (0.718-0.756)	(0.526-0.827)	0.73 (0.04)	0.741 (0.717-0.755)	(0.404-0.823)
	Missing with MAR	0.73 (0.039)	0.743 (0.714-0.756)	(0.576-0.8)	0.728 (0.039)	0.739 (0.713-0.754)	(0.502-0.809)
	Missing with MNAR	0.73 (0.04)	0.743 (0.714-0.755)	(0.57-0.813)	0.728 (0.039)	0.74 (0.715-0.754)	(0.544-0.814)
	Imputed MCAR	0.727 (0.042)	0.74 (0.707-0.755)	(0.539-0.818)	0.727 (0.042)	0.739 (0.71-0.755)	(0.526-0.815)
	Imputed MAR	0.729 (0.041)	0.741 (0.711-0.756)	(0.573-0.821)	0.729 (0.038)	0.74 (0.714-0.754)	(0.525-0.813)
	Imputed MNAR	0.729 (0.038)	0.74 (0.713-0.754)	(0.558-0.811)	0.728 (0.038)	0.739 (0.711-0.753)	(0.596-0.813)
Medium	Original	0.871 (0.029)	0.88 (0.863-0.89)	(0.714-0.923)	0.871 (0.027)	0.88 (0.862-0.889)	(0.727-0.924)
	Missing with MCAR	0.869 (0.03)	0.878 (0.86-0.888)	(0.686-0.923)	0.871 (0.029)	0.879 (0.861-0.89)	(0.618-0.924)
	Missing with MAR	0.87 (0.03)	0.88 (0.863-0.89)	(0.747-0.926)	0.87 (0.028)	0.879 (0.86-0.889)	(0.726-0.922)
	Missing with MNAR	0.871 (0.028)	0.879 (0.862-0.889)	(0.73-0.919)	0.87 (0.028)	0.879 (0.861-0.889)	(0.734-0.927)
	Imputed MCAR	0.871 (0.028)	0.878 (0.862-0.889)	(0.696-0.926)	0.871 (0.028)	0.879 (0.862-0.889)	(0.655-0.92)
	Imputed MAR	0.871 (0.028)	0.879 (0.863-0.889)	(0.644-0.923)	0.871 (0.028)	0.88 (0.864-0.889)	(0.739-0.914)
	Imputed MNAR	0.871 (0.034)	0.88 (0.863-0.888)	(0.243-0.928)	0.87 (0.029)	0.88 (0.861-0.889)	(0.737-0.934)
High	Original	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (1-1)	(0.998-1)
	Missing with MCAR	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (1-1)	(0.999-1)
	Missing with MAR	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (1-1)	(0.999-1)
	Missing with MNAR	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (1-1)	(0.999-1)
	Imputed MCAR	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (0.999-1)	(0.998-1)
	Imputed MAR	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (0.999-1)	(0.998-1)
	Imputed MNAR	1 (<0.001)	1 (1-1)	(0.999-1)	1 (<0.001)	1 (1-1)	(0.998-1)

MCAR: Missing completely at random; MAR: Missing at random; MNAR: Missing not at random.