In Defense of Zero Imputation for Tabular Deep Learning

Mike Van Ness
Stanford University
mvanness@stanford.edu

Madeleine Udell
Stanford University
udell@stanford.edu

Abstract

Missing values are a common problem in many supervised learning contexts. While a wealth of literature exists related to missing value imputation, less literature has focused on the impact of imputation on downstream supervised learning. Recently, impute-then-predict neural networks have been proposed as a powerful solution to this problem, allowing for joint optimization of imputations and predictions. In this paper, we illustrate a somewhat surprising result: multi-layer perceptrons (MLPs) paired with zero imputation perform as well as more powerful deep impute-then-predict models on real-world data. To support this finding, we analyze the results of various deep impute-then-predict models to better understand why they fail to outperform zero imputation. Our analysis sheds light onto the difficulties of imputation in real-world contexts, and highlights the utility of zero imputation for tabular deep learning.

1 Introduction

Missing values are a ubiquitous problem for real-world tabular data problems. For supervised learning tasks, dealing with missing values is paramount, as most supervised learning models cannot naturally handle missing values. In such cases, imputation is a popular approach, where missing values are replaced by estimates using the observed data. Substantial research exists related to imputation, but often focusing on the imputation task itself rather than the impact on the downstream supervised learning objective. Recent research suggests that accurate imputations in terms of reconstructing the missing values are not necessary for optimal predictions, thus suggesting to jointly optimize imputation and prediction to fully search the imputation space [10, 1].

Neural networks have recently gained more popularity as an alternative to tree-based models for tabular supervised learning [12]. However, unlike trees-based models that can more naturally handle missing values [16], there is no consensus on the best way to handle missing values for neural networks. One promising direction which has been explored recently involves using one joint neural network model for both imputation and prediction, thus directly learning the imputations best for prediction [6, 10, 21, 14]. These *deep impute-then-predict* models are more efficient, since they learn imputations and predictions all in one training stage, and also align with impute-then-predict theory that advocates for such joint optimization [10].

Despite the promise of these recently proposed deep impute-then-predict models, this paper presents an alternative and somewhat surprising conclusion: zero imputation often performs at least as well as deep impute-then-predict models, if not better, on real-world supervised learning tasks with missing values. To support this conclusion, we explore why these deep impute-then-predict models cannot beat zero imputation. Our contributions are the following:

- To our knowledge, we are the first to compare multiple deep impute-then-predict pipelines to each other on realistic supervised learning tasks, as well as to zero imputation as a baseline. These

Table Representation Learning Workshop at NeurIPS 2022..

experiments show that impute-then-predict pipelines rarely outperform zero imputation for tabular deep learning.
- We investigate why deep impute-then-predict models fail to beat zero imputation, revealing novel insights into how the imputations of different deep impute-then-predict models differ and when such imputations succeed or fail.

2 Deep Impute-Then-Predict

Deep impute-then-predict models employ one neural network for both imputation of missing values and prediction of supervised labels, constrasting traditional approaches that use separate models for imputation and prediction. There are multiple benefits, at least intuitively, for this deep impute-then-predict setup. For one, using neural networks for both imputation and prediction allows for joint optimization of the imputation and prediction models via backpropagation of the supervised loss. Since the true values for the missing entries are not known, optimizing the imputation network separately is less desirable, since the quality of the imputations cannot be assessed. Further, optimizing imputations and predictions jointly is more efficient, since only one round of optimization is required, and standalone imputations models can be considerably slower than supervised models [17].

2.1 Deep Impute-Then-Predict Models

A deep impute-then-predict model can be created by combining any imputation and prediction neural networks, as long as both models can be optimized via a supervised loss. Nonetheless, some recent papers have presented models specifically designed to be used together for both imputation and prediction. We consider the following impute-then-predict models in this paper:

- **NeuMiss**: a deep impute-then-predict model inspired by the EM algorithm [9]. Under the assumption that the complete data $\boldsymbol{X}$ is multivariate normal, NeuMiss uses a specialized neural network to combine EM-like imputations with supervised predictions. Theory and experiments show that NeuMiss works well for MCAR, MAR, and even certain MNAR data.
- **supMIWAE**: a deep impute-then-predict model that use a Variational Autoencoder (VAE) for multiple imputation [6]. Instead of a normal supervised loss, supMIWAE optimizes an evidence lower bound objective for the entire pipeline, including imputations and predictions.
- **GRAPE**: a deep impute-then-predict model based on graph neural networks [21]. A bipartite graph is built from the features and samples of a tabular dataset, with edges only for observed feature-sample pairs.

For each model, we also consider using the missing indicator method (MIM) to capture any informativeness in the missing values [17]. For MIM with SupMIWAE, we combine SupMIWAE and NotMIWAE [5], which we call SupNotMIWAE, to incorporate MIM into the loss function, see Appendix D for more details.

2.2 Other Related Work

Despite significant research on imputation, including using deep learning [3, 20, 11], much less previous work has focused on the impact of imputation on supervised learning. Other than the aforementioned deep impute-then-predict papers, some papers on decision trees [16, 7] and some empirical studies [2, 19] have studied the impact of imputation on supervised learning. Additionally, some previous work has advocated for simple alternatives like zero imputation. For one, the missing indicator method often pairs with zero imputation [17]. Specific to neural networks, some recent tabular deep learning models propose using zero imputation to handle missing values [15, 4].

3 Experiments

3.1 Setup

We empirically evaluate the deep impute-then-predict models discussed in Section 2.1 compared to the baseline of an MLP model with zero imputation. Since all 3 deep impute-then-predict models we consider use MLPs for prediction, all 4 methods (including zero imputation) we consider differ

Figure 1: Prediction and imputation scores (lower is better) for deep impute-then-predict models on various OpenML datasets, with and without MIM. Datasets without results for GRAPE were too large for GRAPE to terminate.

in architecture only in their imputation networks. We select several datasets from OpenML [18] that initially have no missing values for evaluation (see Table 1 in Appendix A). For each dataset, we generate missing values using a missing completely at random (MCAR) missing mechanism (for results on missing not at random (MNAR) data, see Appendix B). For each dataset and model combination, we run 5 trials with different random seeds as well as different missing masks. We report 1 - AUC for binary classification tasks, 1 - accuracy for multiclass classification tasks, and RMSE for imputation error and regression tasks (so lower is always better). Additional experiment details are available in Appendix A, and code to reproduce the results is available [1].

3.2 Main Results

The top panel of Figure 1 shows the performance of the selected deep impute-then-predict models and zero imputation (MLP) on the OpenML datasets with MCAR missing values. Even though non-zero imputation is usually most effective on MCAR data, zero imputation has very competitive performance compared to the deep impute-then-predict models. These results illustrate the utility of zero imputation, especially since it is more simple and more efficient than the deep impute-then-predict models, and it allows imputed values to be ignored by gradient updates (see Appendix C for details).

3.3 Analysis

Why does zero imputation perform as well as deep impute-then-predict models? On the one hand, theory [1, 10] suggests that any imputation can result in a Bayes optimal model given a powerful enough prediction function. On the other hand, finding such a powerful prediction function in practice is challenging if not impossible, and thus it is reasonable to think jointly optimizing imputation and prediction would yield the best performance.

What imputations do deep impute-then-predict models produce? Do deep impute-then-predict models try to reconstruct the missing values? If not, do they make imputations that follow some noticeable pattern? Answers to these questions do not appear in previous literature, including in papers that propose deep impute-then-predict models. The bottom panel in Figure 1 shows the imputation RMSEs for each model. In many cases, the deep impute-then-predict models do not reconstruct the missing values more accurately than zero imputation. This is somewhat surprising, since reconstructing the missing values is easier with MCAR missing values, yet these models either cannot find such imputations with lower imputation RMSE, or do not think they are better. Further, Figure 2 shows that the deep impute-then-predict models (except GRAPE) tend to find imputations with Gaussian-like distributions, despite the true values being clearly non-Gaussian (see Figure 5

[1] `https://anonymous.4open.science/r/deep-impute-then-predict-2712/README.md`

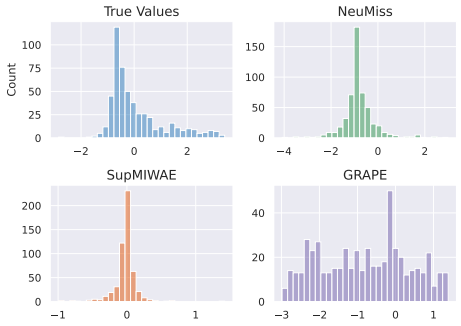

Figure 2: True distribution of missing values (upper left) compared to distributions of imputed values by deep impute-then-predict models on the first feature of the phoneme dataset.

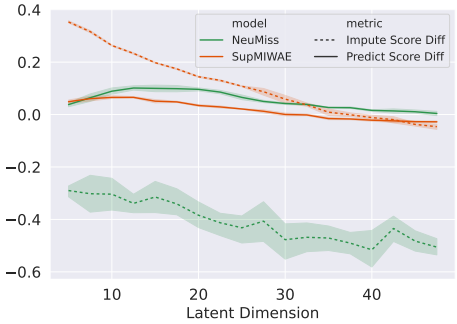

Figure 3: Improvement in imputation and prediction RMSE compared to zero imputation on synthetic regression data as a function of data latent dimension. A smaller latent dimension corresponds to an easier imputation task.

in the Appendix for more imputation distribution plots). The NeuMiss imputations additionally are centered below 0, indicating a bias towards imputing under the true values.

When is it possible to beat zero imputation? When deep impute-then-predict models can outperform zero imputation on synthetic data (as in [5, 10]), what is unique about the synthetic data, and what imputations do the models produce? To investigate, we run NeuMiss and SupMIWAE on synthetic data generating as in [9] (we exclude GRAPE due to inefficiency), using the latent dimension (rank) of the data covariance matrix as a proxy for imputation difficulty, with a smaller latent dimension resulting in imputation easier. The results are shown in Figure 3. When imputation is easier (smaller latent dimension), the deep impute-then-predict models are able to significantly outperform zero imputation, as expected. When imputation is harder (larger latent dimension), however, SupMIWAE is much worse than zero imputation, while NeuMiss performs comparably to zero imputation (likely because the assumptions for NeuMiss are met on the synthetic data). This result suggests that the deep impute-then-predict models struggle to beat zero imputation in Figure 1 because imputation is too challenging on real-world data, even with MCAR missingness. Looking at the imputation reconstruction errors is also insightful: SupMIWAE finds imputations that try to reconstruct the missing values, while NeuMiss does not, yet both are successful with sufficiently small latent dimension. This serves as an example that imputations with large RMSE can outperform zero imputation, but only when imputation is sufficiently easy.

4 Conclusion

We present an analysis of various strategies for neural networks to manage missing values in tabular supervised learning. A very useful finding emerges: zero imputation performs as effectively as, if not better than, complex deep impute-then-predict models in terms of supervised loss. This observation underscores the utility of zero imputation as a easy, reliable, and efficient strategy for tabular deep learning. Furthermore, by analyzing the imputations discovered by deep impute-then-predict models, we find that these models vary in what imputations they produce, and can outperform zero imputation only when imputation is sufficiently easy. We hope that this paper guides practitioners in treating missing values in tabular deep learning, and promotes researchers to consider strategies other than standard deep impute-then-predict models for future papers on missing values.

References

[1] D. Bertsimas, A. Delarue, and J. Pauphilet. Prediction with missing data. *stat*, 1050:7, 2021.

[2] U. Garciarena and R. Santana. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications*, 89:52–65, 2017.

[3] L. Gondara and K. Wang. Mida: Multiple imputation using denoising autoencoders. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*, pages 260–272. Springer, 2018.

[4] N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2022.

[5] N. B. Ipsen, P.-A. Mattei, and J. Frellsen. not-miwae: Deep generative modelling with missing not at random data. *arXiv preprint arXiv:2006.12871*, 2020.

[6] N. B. Ipsen, P.-A. Mattei, and J. Frellsen. How to deal with missing data in supervised deep learning? In *ICLR 2022-10th International Conference on Learning Representations*, 2022.

[7] J. Josse, N. Prost, E. Scornet, and G. Varoquaux. On the consistency of supervised learning with missing values. *arXiv preprint arXiv:1902.06931*, 2019.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] M. Le Morvan, J. Josse, T. Moreau, E. Scornet, and G. Varoquaux. Neumiss networks: differentiable programming for supervised learning with missing values. *Advances in Neural Information Processing Systems*, 33:5980–5990, 2020.

[10] M. Le Morvan, J. Josse, E. Scornet, and G. Varoquaux. What's a good imputation to predict with missing values? *Advances in Neural Information Processing Systems*, 34, 2021.

[11] P.-A. Mattei and J. Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning*, pages 4413–4423. PMLR, 2019.

[12] D. McElfresh, S. Khandagale, J. Valverde, G. Ramakrishnan, M. Goldblum, C. White, et al. When do neural nets outperform boosted trees on tabular data? *arXiv preprint arXiv:2305.02997*, 2023.

[13] A. Perez-Lebel, G. Varoquaux, M. Le Morvan, J. Josse, and J.-B. Poline. Benchmarking missing-values approaches for predictive models on health databases. *GigaScience*, 11, 2022.

[14] M. Śmieja, Ł. Struski, J. Tabor, B. Zieliński, and P. Spurek. Processing of missing data by neural networks. *Advances in neural information processing systems*, 31, 2018.

[15] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

[16] B. E. Twala, M. Jones, and D. J. Hand. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters*, 29(7):950–956, 2008.

[17] M. Van Ness, T. M. Bosschieter, R. Halpin-Gregorio, and M. Udell. The missing indicator method: From low to high dimensions. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5004–5015, 2023.

[18] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

[19] K. Woźnica and P. Biecek. Does imputation matter? benchmark for predictive models. *arXiv preprint arXiv:2007.02837*, 2020.

[20] J. Yoon, J. Jordon, and M. Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pages 5689–5698. PMLR, 2018.

[21] J. You, X. Ma, Y. Ding, M. J. Kochenderfer, and J. Leskovec. Handling missing data with graph representation learning. *Advances in Neural Information Processing Systems*, 33:19075–19087, 2020.

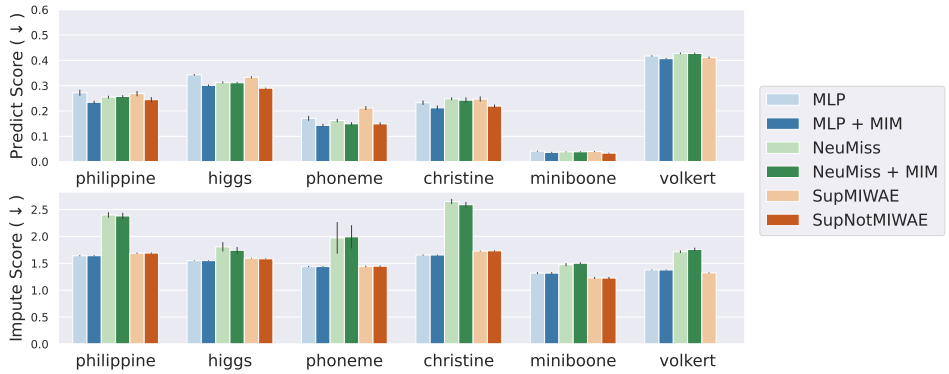

Figure 4: Results parallel to Figure 1 but for data with MNAR missing values.

A Additional Experiments Details

For all datasets, we do a 60/20/20 train/validation/test split. All features are standardized by subtracting by the mean and dividing by the standard deviation over the observed values on the training data. Note that zero imputation is equivalent to mean imputation after standardization. We run 5 trials and average the results for all experiments. Error bars in each result are the standard errors of the mean. We use the Adam optimizer [8] with a learning rate of 1e-4. We train each model for a maximum of 100 epochs, and early stop any trials that do not improve performance on the validation set for 5 consecutive epochs.

To generate missing values, for MCAR we mask each feature value with a probability of 0.5. For MNAR missing values (see Appendix B), we generate missing values following [17] with an informativeness parameter γ of 2. Once values are masked, we only use the underlying true values to calculate imputation RMSE, not for any part of training.

Lastly, Table 1 below describes the OpenML datasets used for our experiments.

Table 1: OpenML data sets used.

OpenML ID	Name	n	p	Task	n_classes
23512	higgs	98050	28	binary	2
41150	miniboone	130064	50	binary	2
41145	philippine	5832	309	binary	2
41142	christine	5418	1599	binary	2
1489	phoneme	5404	5	binary	2
41166	volkert	58310	147	multiclass	10

B Results on MNAR Data

The results in Section 3 focus on data with MCAR missing values. However, real-world data often has missing values that follow some MNAR mechanism, e.g. informative missingness in healthcare data [17, 13]. Thus, we also run experiments to analyze the performance of deep impute-then-predict models on MNAR data. We generate MNAR missing values following [17] using an informativeness parameter γ of 2. The results are shown in Figure 4. Similarly to the MCAR case, MLP with zero imputation performs as well as the deep impute-then-predict models. Unlike the MCAR case, however, the missing indicator method boosts performance almost all models, and MLP with MIM achieves as good performance as any other method. This supports the conclusions in [17] that zero imputation with MIM is a generally strong way to handle missing values when optimizing for supervised learning.

C Zero imputation in Neural Networks

Mean imputation (or zero imputation are standardization) is perhaps the most common approach for imputing missing values across all supervised learning models. For neural networks, though, zero imputation is somewhat unique amongst imputation approaches. Consider the first layer of a neural network as a linear layer

$$h = Wx + b = \sum_i w_i x_i + b, \quad W = [w_1 \quad w_2 \quad \cdots \quad w_p] \tag{1}$$

for input $x \in \mathbb{R}^p$, weight $W \in \mathbb{R}^{d \times p}$, and bias $b \in \mathbb{R}^d$. If x_i is imputted with 0, then the corresponding terms disappears in the sum in Equation 1. Further, the derivative of h with respect to w_i is

$$\frac{\partial h}{\partial w_i} = x_i \mathcal{I} \tag{2}$$

where $\mathcal{I}$ is the $d \times d$ identify matrix. Thus, if x_i is imputed with 0, then $\partial h / \partial w_i = 0$ (where 0 here means the $d \times d$ matrix of all zeroes). Since the derivative is 0, assuming standard gradient decent, w_i will receive no update during backpropagation. Therefore, the weights connected to the missing entry are not updated when the missing entry is imputed with 0. In other words, the network only updates the weights in the first layer corresponding to the entries that are observes, allowing the missing entries to be ignored in a sense. This is desirable behavior, and may partially explain why zero imputation achieves pretty strong performance in our experiments.

D MIM with SupMIWAE

For NeuMiss and GRAPE, it is straightforward to incorporate MIM by concatenating the indicator features to the output of the imputation network. With SupMIWAE, however, the imputation and prediction networks are optimized via an evidence lower bound (ELBO) that needs to be adjusted to incorporate MIM. Let y be a supervised response and x a vector of features, partitioned into observed parts x_o and missing parts x_m, and r the vector of missing indicators. SupMIWAE optimizes the observed log-likelihood $\log p_{\phi,\theta,\psi}(y, x_o, r)$ assuming the latent variable model

$$\log p_{\phi,\theta}(Y, X_o) = \log \int p_\phi(Y \mid X_o, X_m) p_\theta(X_o, X_m \mid Z) p_\theta(Z) dZ dX_m. \tag{3}$$

This log-likelihood can be maximized by maximizing the ELBO:

$$ELBO_{supmiwae} = \mathbb{E}_{p_\theta(X_m|Z), q_\gamma(Z|X_o)} \left[\log \frac{1}{K} \sum_{i=1}^{K} \frac{p_\phi(Y \mid X_o, X_m^i) p_\theta(X_o \mid Z^i) p(Z^i)}{q_\gamma(Z^i \mid X_o)} \right], \tag{4}$$

see [6] for more details. To incorporate MIM into this ELBO, instead of maximizing the observed log-likelihood in Equation 3, we have to maximize the full log-likelihood

$$\begin{aligned} &\log p_{\phi,\theta,\psi}(Y, X_o, R) \\ &\qquad = \log \int p_\phi(Y \mid X_o, X_m, R) p_\psi(R \mid X_o, X_m) p_\theta(X_o, X_m \mid Z) p_\theta(Z) \, dZ \, dX_m \end{aligned} \tag{5}$$

where $p_\phi(Y \mid X_o, X_m, R)$ represents the prediction neural network that uses MIM. The corresponding ELBO to maximize is

$$\begin{aligned} &ELBO_{sup-notmiwae} \\ &\quad = \mathbb{E}_{p_\theta(X_m|Z), q_\gamma(Z|X_o)} \left[\log \frac{1}{K} \sum_{i=1}^{K} \frac{p_\phi(Y \mid X_o, X_m^i, R) p_\psi(R \mid X_o, X_m) p_\theta(X_o \mid Z^i) p(Z^i)}{q_\gamma(Z^i \mid X_o)} \right]. \end{aligned}$$

This combines the ideas from SupMIWAE (MIWAE for prediction) and NotMIWAE (MIWAE for MNAR data) [5].

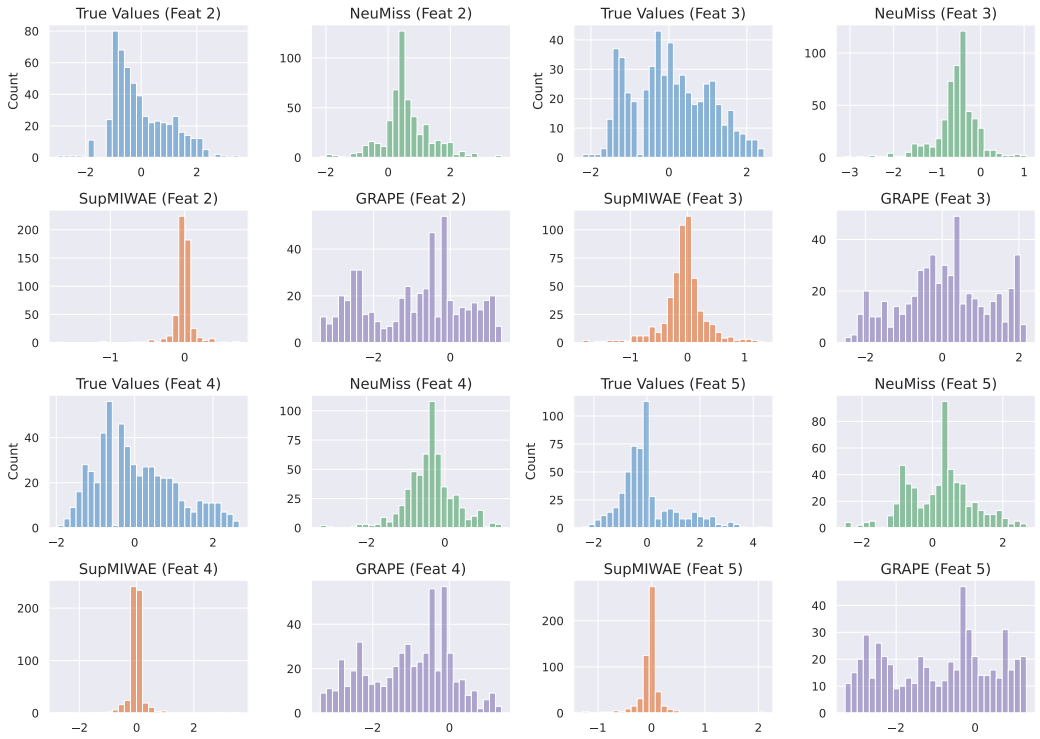

Figure 5: Parallel figure to Figure 2 for the remaining features in the phoneme datasets. The trends from Figure 2 are similar in the remaining features.