FILLING MISSING VALUES FOR AI-BASED (LOAD) FORECASTS WITHIN THE INTERFLEX MICRO GRID DEMO IN SIMRIS, SWEDEN

Roxana POHLMANN
RWTH Aachen – GER
roxana.pohlmann@rwth-aachen.de

Henning WILMS
RWTH Aachen – GER
hwilms@eonerc.rwth-aachen.de

Marco CUPELLI
RWTH Aachen – GER

Inko ELGEZUA FERNANDEZ
E.ON – GER

Antonello MONTI
RWTH Aachen – GER

ABSTRACT

Missing data impairs the performance of most neural networks with a particularly strong effect on time series prediction networks. Imputation addresses this issue and by replacing missing values with substitute values. The choice of a suitable imputation method requires fundamental knowledge of the dataset.
Autoencoders (AE) have been widely applied in representation learning and feature extraction. In this paper we use a stacked denoising overcomplete autoencoder for imputation in multi-variate time series.
We assess the model's feature reproduction capability and compare its effect to simple mean imputation on a open source data set. Moreover, we assess the imputation's influence on a recurrent neural network's short-term load forecasting results and show that our proposed autoencoder model yields better results in feature imputation and significantly improves the forecasting accuracy for low and high fractions of missing data.

INTRODUCTION

Due to the energy transition, increasingly more renewable and distributed energy resources become part of our power systems. These energy sources add volatility and stochasticity to the power system, making grid operation increasingly difficult. Short term load forecasting (up to two weeks) hence is an essential input for grid operation of power systems with high share or renewable or distributed energy sources [3, 11].
The control of a decentralized energy grid requires reliable locally adapted solutions. Neural networks (NN) are one technique for load time series forecasting. These algorithms extrapolate the future behavior from learned patterns, which they derive from large historical datasets. In these datasets, any exogenous input variables are called features and are used to infer the correct target value, called label [3, 10, 11].
Training forecasting networks with real-world datasets seems desirable, but the existence of outliers and missing data can decrease or even manipulate the learning process. Statistics considers three different kinds of missing data mechanisms:

- Data sets with data *missingness completely at random* (MCAR) show the same probability for missingness for all instances.
- With *missingness at random* (MAR) the missingness of a variable depends on other available information.
- *Missingness not at random* (MNAR) applies if missing values depend on the unobserved information or on the value itself.

MNAR must be modelled explicitly, otherwise it will cause bias, whereas MAR and MNAR are considered as ignorable [4].
The existence of missing values can motivate the user to narrow its dataset by dropping a feature or excluding parts of a dataset. Therefore, an effective handling of missing data is important and poses the question how to select an appropriate model for value replacement without manipulating the data structure itself. This task is called *imputation*.
Simple imputation approaches include dropping instances with missing entries or replacing the entries with the feature's mean (mean imputation). While all algorithms suffer from missing data, dropping instances has severe effects for time series forecasts. Time series forecasting algorithms rely on capturing time series dynamics. Dropping missing values may lead to discontinuous time steps or a varying length of time steps [10].
Mean imputation disturbs the distribution of the dataset in a way that does not account for the sensibility of NNs. The risk being that they learn false relationships, which causes a delayed or disturbed learning process. In this work we will focus on imputing missing data in time series for short term load forecasting (STLF) that use recurrent neural networks (RNNs) as predictors [2].
The proposed architecture combines a denoising auto-encoder (DAE) for imputation with a predictive RNN. The DAE captures the underlying structure of a dataset; thereby it simultaneously imputes missing entries and extracts features from the input dataset. AEs' popularity stems from their ability to filter useful features from a dataset in an unsupervised manner [8, 9]. Bengio showed that AEs do not only capture information about the structure in a dataset, they even implicitly recover the data generating distribution of a dataset [1], which both are relevant properties we aim to exploit for the imputation task at hand.
Our model follows up *Gondara's* architectural proposition of an overcomplete DAE [5], with a focus on imputing times series for STLF. In this paper, we show that DAEs are able to impute missing values in temperature time series data in a useful way. Moreover, the DAE imputed time series considerably increase the accuracy of the load forecasting in comparison to mean-imputed time series for up to 20% of missing input data. We evaluate these

questions using the open source available Global Energy Forecasting Competition (GEFCom) load data from 2012 [7], which comprises temperature readings as features to infer load forecasts.

BACKGROUND

Artificial Neural Networks

(Artificial) neural networks (NNs) consist of neurons connected to each other by weighted connections. The layers between inputs and outputs are called hidden layers (with hidden neurons) and define the depth of an NN [6].
With NNs, an important distinction is between variables and hyperparameters. Variables include weights and biases, which iteratively change during a training session until the best solution is found. During such a training session, the variables are adapted in direction of minimizing the objective function (e.g. forecasting error, reconstruction error). In contrast, hyperparameters specify the architecture and training configuration of the network, e.g. the number of hidden layers, the number of neurons per hidden layer, or the learning rate [6].
A feedforward NN transmits information only from the input in direction of the output. In contrast, RNNs additionally possesses cyclic connection; thereby an RNN can capture the dynamic behavior of time sequences [11].

Autoencoder

An AE is a feedforward NN the same number of neurons in the input and output layer. Their goal is to learn a useful representation of a dataset in a lower-dimensional space following the manifold assumption by encoding the input into the hidden representation (codings) and decoding it back [1]. As the reconstruction's quality is measured as distance between the input (x) and the output ($\tilde{y}$) measures the reconstruction's quality, AEs aim at minimizing the mean squared error (MSE)

$$L(x, \tilde{y}) = \sum_{i=1}^{n} (\tilde{y}_i - x_i)^2.$$

The underlying idea for this kind of training assumes that the codings must contain the relevant information in the data, since the output layer builds up only from the codings whilst still minimizing the error between input and output [1]. Classical AEs are undercomplete with the number of neurons in the hidden layer remaining smaller than the number of inputs in order to find lower dimensional representations. In overcomplete representations the number of neurons in the hidden layer is larger than in the input layer. Classical unregularized AEs do not allow overcomplete representations since this would allow duplicating the input features to the output. For the imputation task at hand an overcomplete hidden layer is implemented, as it allows for mapping the input data to a higher dimensional space and thus enables an easier reconstruction of missing values. This architecture can be realized by using a DAE, a modification of the AE [8].

METHODOLOGY

Denoising Autoencoder

DAEs attempt to learn a robust lower-dimensional manifold of the training dataset by corrupting a part of their input during training with the goal of recovering the original input, irrespective of the corruption (Fig. 1) [8]. One method of corruption is the application of a masking noise, which sets a fraction *v* of the input data to zero.
Stochastically the corrupted data parts lie further apart from the underlying distribution than the uncorrupted data. Thus, the AE learns to map the corrupted and uncorrupted data back on the manifold and ignores noises, which could be missing entries or outliers in the data. In contrast to classical AEs, overcomplete DAEs with a larger hidden layer than input layer have a small risk of learning the identity mapping [9].

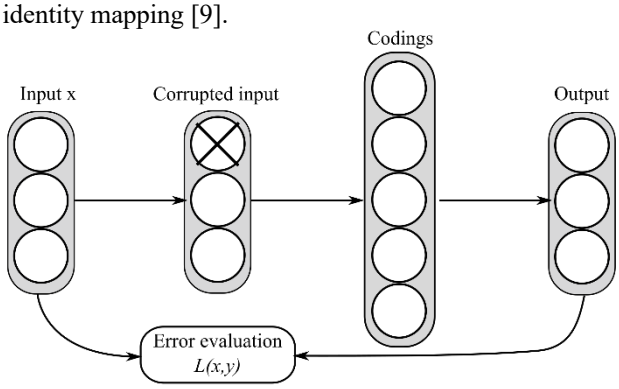

Fig. 1 A DAE learns to map a corrupted input back to the original values.

Autoencoder Recurrent Neural Network Model

The proposed AE-RNN consist of an AE whose results are postprocessed by an RNN with the RNN yielding the forecast (Fig. 2). An overcomplete stacked DAE model is employed for the AE, as an increase of data dimensionality and variance is desired for the required imputation task. The DAE applies a dropout noise at fraction *v* setting random inputs to zero during training. This dropout noise is not applied during inference.
In a real-world scenario, the user will only have access to a dataset with preexisting missing values. The proposed model considers this challenge through the selected denoising approach. As a neural network cannot handle absent not a number (NaN) entries, missing values are replaced with the feature's mean (standard mean imputation). The DAE trains while applying dropout noise on part of the dataset learning to map data to the overcomplete manifold even if random values are set to zero (mimicking missing values).

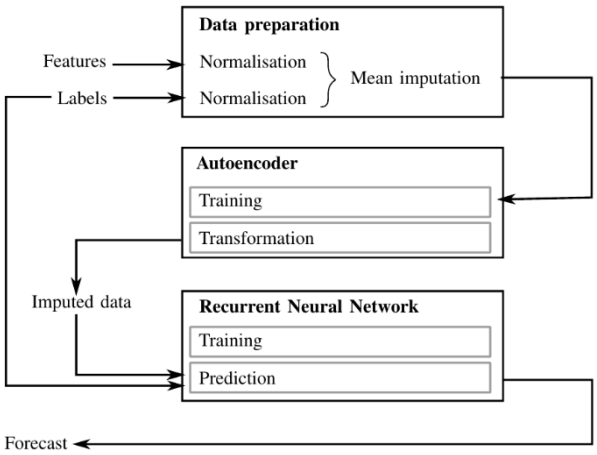

Fig. 2: The AE-RNN model requires a normalized mean-imputed dataset. The AE is trained with great part of the dataset until convergence. Then it transforms the features and passes them to the RNN.

After the training is completed, the DAE transforms the complete dataset without applying dropout noise and reconstructs missing values. This transformed data will be referred as AE imputed data, the quality of the reconstruction depends on the DAE's data representation's quality.

The hyperparameters for the considered DAE include the learning rate, the fraction of corruption (v), the number of neurons and layers and the activation function.

The RNNs most important hyperparameters are the number of layers and the number of neurons per layer, the learning rate, the bidirectionality of the network, and the cell type.

The networks are not connected with each other; instead, each network possesses its own variables and is trained individually. A combined training would drastically increase the degrees of freedom in the system and subsequently increase the training time. Moreover, a connected network might result in specialized solutions for weights and biases not guaranteeing that each part individually (DAE or RNN) delivers satisfying results. By training both models separately we can assure that each of the models performs and is optimized for their respectively desired task (reconstruction or forecasting). Random Search (RS) selects the hyperparameters for both network types [6].

EXPERIMENTAL SETUP

Evaluation pipeline

The performance and behavior of neural networks highly depends on the selected hyperparameters and the dataset used for training and evaluation. Therefore, the effect of missing data will be simulated by intentionally disturbing the temperature data with varying fractions of missing entries μ. RS delivers the hyperparameters of the RNN on the complete dataset. These hyperparameters remain fixed for all μ. This procedure ensures comparability between the RNN taken as reference and the AE-RNN models as well as comparability between the performance with disturbed and complete data.

Fig. 3 visualizes the process of evaluation for every model and every fraction of corruption. In the first step, the RNN is trained with the mean imputed data. The evaluation of this model (evaluation 1) serves as reference for the AE-RNN models. Equally, the AE is trained with the mean imputed data until a break criterion is reached, i.e. sufficient accuracy.

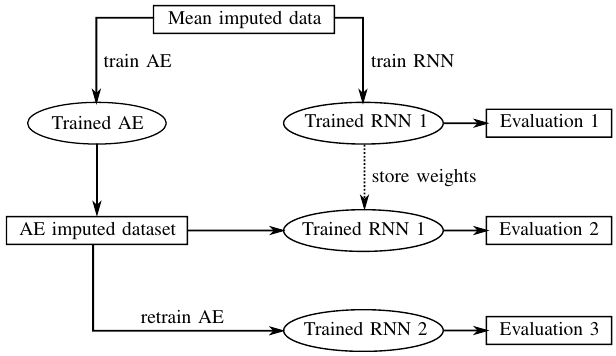

Fig. 3: The evaluation scheme of the combined network.

Secondly, the stored and trained RNN 1 is evaluated with the AE imputed data (evaluation 2). Evaluation 2 indicates changes in the forecast, which are probably caused by differences between the mean imputed data and AE imputed data. Finally, the RNN is retrained with the AE imputed data (trained RNN 2) and again evaluated (evaluation 3).

While the RNN's hyperparameters are fixed, RS determines two DAE candidates for each μ: the first AE candidate delivers the best reconstruction measured between input and output layer, it will be references as "on reconstruction" (OR). The second AE candidate reaches the lowest prediction error of the pretrained RNN (evaluation 2), it will be references as "on prediction" (OP).

Evaluation metrics

Evaluate imputation

We evaluated the model using data with different fractions of missing data, thus we used the original and complete data as reference for good imputation. Three datasets are proposed for the evaluation, the original complete dataset (D_{org}), the intentionally corrupted dataset (D_{cor}) and the AE imputation result (D_{imp}).

The imputation score measures the change of the distances between the corrupted and imputed data relative to the distance between corrupted and original data as

$$\theta_{RMSE} = 1 - \frac{RMSE(D_{org}, D_{imp})}{RMSE(D_{org}, D_{cor})}.$$

A large positive θ_{RMSE} indicates a good imputation while a negative θ_{RMSE} accounts for an increased distance to the original dataset. The score takes the overall error on the dataset into account, but not the existence of outliers.

Evaluate prediction

The prediction success of the combined model is measured as ratio between the prediction error of the AE-RNN and the prediction error of the RNN with mean imputation, both obtained with the same missing data entries, as

$$\Gamma(\mu) = \frac{MSE_{AE-RNN}(\mu)}{MSE_{RNN}(\mu)}.$$

Here, an improvement of the forecasting accuracy by the combined model compared to the mean imputed RNN yields $\Gamma < 1$ whereas $\Gamma > 1$ indicates a deterioration.

Configuration

The dataset contains 15 features. Therefore, we assume that DAE architectures using one hidden layer or three hidden layers are deep enough for feature extraction. The RS parameter spaces iterate over the number of neurons, the learning rate, the dropout fraction (v) and the activation function.

Dataset

We train and evaluate the AE-RNN network using the GEFCom load dataset [7]. This dataset contains 4.5 years of hourly load and temperature measurements with the corresponding timestamps (including year, month, day, hour) from 2008/6/30 to 2014/1/1 for 11 different geographical zones. The goal is to forecast the load profile for 20 different nodes within the 11 geographical zones. The real-world GEFCom dataset includes errors like outages and load transfers. The percentage of NaN entries in the original temperature dataset is 0.034%. Like [5], further missing data will be simulated by deleting random entries in the feature set. Regarding the temperature dataset (the input features), all temperature sensors are considered to have the same probability of outage independent of their location, the date, the time, and the measured temperature. Thus, MCAR is assumed which is modelled by randomly setting entries in the temperature data to NaN with fractions of $\mu \in \{0.5\%, 1\%, 2.5\%, 5\%, 7.5\%, 10\%, 15\%, 20\%, 30\%, 40\%, \text{and } 50\%\}$. The MSE of solely the RNN rapidly increases already for small fractions of corruptions, i.e. $\text{MSE}(\mu = 0.5\%) = 2.23$, $\text{MSE}(\mu = 5\%) = 2.74$.

RESULTS

The AE-RNN's evaluation consists of two parts: the assessment of the imputed data and the performance comparison to an RNN with only mean imputation. The reader may note the following abbreviations indicating different architectures:

- 1 layer / 1L: DAE with one hidden layer
- 3 layers / 3L: stacked DAE with three hidden layers
- On reconstruction (OR) scored: the RS score is the MSE calculated on the AE's input and reconstruction.
- On prediction (OP) scored: the RS score is the prediction MSE of the trained RNN.

Imputation Performance

The imputation performance is measured as θ_{RMSE}. The evaluation benchmark shows the best values for small fractions of corruption and vanishes for corruption fractions larger than $\mu = 0.1$. Tab. 1 shows selected values of the imputation performance in the most relevant section of $\mu \leq 0.1$.

Architecture	*score*	$\mu = 0.5\%$	$\mu = 2.5\%$	$\mu = 10\%$
1 layer	OR	3.82	1.35	0.43
	OP	5.95	1.94	0.71
3 layers	OR	4.94	2.62	1.44
	OP	0.82	17.27	14.96

Tab. 1: The evaluation of the imputation performance measured as $\theta_{RMSE} * 10^3$.

Prediction Performance

The performance of the combined models' prediction is measured by $\Gamma_{XL-OY}(\mu)$, where X is the number of hidden layers of the DAE (1 or 3), Y the score type (on prediction - OP or on representation - OR) and μ again is the fraction of corrupted data.

The dashed lines in Fig. 4 indicate the error of the standard RNN without the DAE imputation as reference value (black, 1), the error of the OP model before (2a) and after training (2b) and the error of the OR model before (3a) and after training (3b).

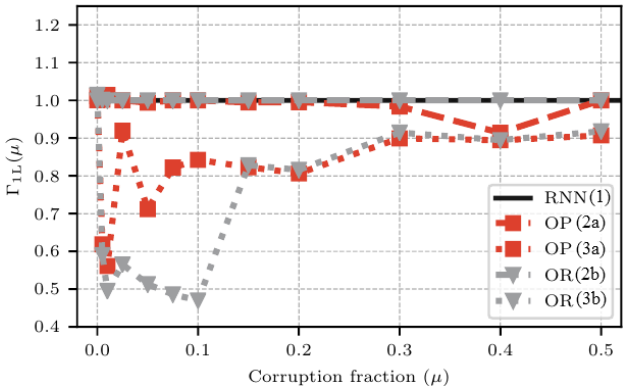

Fig. 4: The prediction score of the combined model $\Gamma(\mu)$.

As the lines 2a and 3a approach or intersect the reference line, the AE imputed data (evaluation 2) does not significantly improve the forecast of the pretrained model. After retraining the variables, a clear improvement of prediction accuracy is visible for both architectures (2b, 3b). The OR scored model (3b) outperforms the OP scored model (2b) until and including a corruption faction of 0.1, afterwards the prediction accuracy of both models approaches and converges. None of the models is capable to reduce the prediction error on the complete dataset. Other error measures show identical trends.

Discussion

This work addresses the problem of missing entries in datasets with a special focus on time series forecasting for STLF using RNNs. Installing a DAE before the RNN aims at making these corrupted features accessible to the RNN. We propose an overcomplete DAE being a useful imputation method to recover time series datasets with missing entries, since a DAE can capture the data

generating distribution of a dataset.
Our results show the DAE working well in recovering a small to significant part of missing and mean imputed values (Tab. 1). The effect is strong for small corruption fractions and vanishes for larger amounts of missing data. The correlation can be explained, as the progressive destruction of data causes a loss of significant data characteristics, which the AE does not have the chance to learn and revert. The models with three hidden layers outperformed the models with one hidden layer. Overall, the three-layer DAEs delivered lower imputation errors, which were increasing less quickly. This does seem to depend on the deeper architecture with an improved generalization capability. Moreover, the best models all possess a significant high number of neurons which substantiates the approach of using overcomplete DAEs for imputation.
When faced with corrupted data, the combined AE-RNN model's forecast is substantially better than the standard RNN (Fig.4). It is notable that the OP scored models outperform the OR scored ones in imputation and pretraining forecasts (evaluation 1 and 2), whereas they are outperformed in retrained combined models for small corruption fractions (evaluation 3).
Interestingly, our results demonstrated that even models with a low imputation performance, succeeded in clearly improving the overall prediction. The feature extraction capability of the DAE seems to cause this effect. However, our model did not succeed in improving the performance on a complete dataset in comparison to the original RNN. We speculate that this might be due to the use of the AE's reconstruction instead of using lower-dimensional codings.

CONCLUSION

To conclude, the DAEs clearly succeeded in recovering mean imputed values by learning the data distribution for small fractions of missing entries. The effect vanishes with an increasing percentage of missing data, although a small improvement was observed for all models and corruption fractions. Moreover, the combined AE-RNN models predictions achieve a substantially better accuracy than the RNN's prediction with an incomplete mean imputed dataset.
DAEs offer a robust imputation method with only few necessary design choices. Therefore, the method serves as option for automatic data pre-processing with the advantages of computationally inexpensive training and an implicit learning of the data-generating distribution.
In this work, we show the effectiveness of this method. In future work the performance of DAEs' imputation results should also be compared to imputation methods different from mean imputation.

REFERENCES

[1] Bengio, Y and Courville, Aaron and Vincent, Pascal. 2013. Representation Learning: A Review and New Perspectives. *IEEE transactions on pattern analysis and machine intelligence,* 35, 1798–1828.

[2] Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., A, R., and Jenssen, R. 2017. *Recurrent Neural Networks for Short-Term Load Forecasting. An Overview and Comparative Analysis*. SpringerBriefs in Computer Science. Springer International Publishing, Luxembourg.

[3] Bogdanovic, M., Wilms, H., Cupelli, M., Hirst, M., & Hernández, L. 2018. Interflex-Simris-Technical management of a grid-connected microgrid that can run in an islanded mode with 100% renewable generation. *In Proceedings CIRED Workshop*, 1–4.

[4] Gelman, A. and Hill, J. 2006. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

[5] Gondara, L. and Wang, K. 2018. MIDA. Multiple Imputation Using Denoising Autoencoders. In *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference*, 260–272.

[6] Goodfellow, I., Bengio, Y., and Courville, A. 2016. *Deep Learning*. MIT Press.

[7] Hong, T., Pinson, P., and Fan, S. 2014. Global Energy Forecasting Competition 2012. *International Journal of Forecasting* 30, 2, 357–363.

[8] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, 1096–1103.

[9] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. 2010. Stacked Denoising Autoencoders. Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* 11, 3371–3408.

[10] Wilms, H., Cupelli, M, Monti, A. 2018. On the Necessity of Exogenous Variables for Load, PV and Wind Day-Ahead Forecasts using Recurrent Neural Networks. *IEEE Electrical Power and Energy Conference (IEEE EPEC)*.

[11] Zheng, J., Xu, C., Zhang, Z., and Li, X. 2017. Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network. In *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, 1–6.