



Fusion of energy sensors with missing values

Amedeo Buonanno¹ · Giovanni Di Gennaro² · Giorgio Graditi³ · Antonio Nogarotto² · Francesco A N Palmieri² · Maria Valenti¹

Accepted: 2 June 2023 / Published online: 13 July 2023
© The Author(s) 2023

Abstract

In Smart Energy Grids, the information flow used to make decisions is the result of fusion of different sources. Communication latency, possible sensor faults and inaccuracies, may negatively impact the data quality and hence the taken decisions. For these reasons, the construction of a robust representation of the input signals that replaces and/or corrects the inaccurate data is crucial for effective classification, anomaly detection and planning. Recent works on Data Fusion and data imputation suggest that the usage of other signals in the same context can empower the representation and can be a useful preprocessing task. In this work we describe an Autoencoder-based data fusion architecture with convolutional layers, skip connections and ad-hoc augmented training sets for data imputation applied to the power consumption measurements obtained by different sub-meters. Among the investigated architectures, the approach with the shared convolutional layers and an augmented dataset that consider missing data in the random positions and located in the central part (AE-A-ALL-CNN), is the most promising one. In presence of one half of the input signal, in the central part, completely erased, it improves the imputation capability, respect to two most employed approaches (denoising autoencoder and MICE) in the average of 12 %.

Keywords Machine learning · Autoencoder · Data fusion · Missing data imputation · Smart grid

1 Introduction

We live in the Data Age, with many and different sensors distributed all around us that measure different aspects of our life. In this scenario efficient data fusion techniques are becoming essential to improve our decision capacity in several contexts [1, 2]. Unfortunately, the real world is all but perfect and many measurements can be missed or corrupted because of communication latency, line interruptions, sensor malfunctioning, etc. In Smart Energy Grids various kinds of sources, such as smart meters, sensors, etc., are used by Smart Energy Managers to make decisions, but the errors in the measure systems or in the communication are frequent [3]. The consequential missing or corrupted measurements impact negatively on the quality of the available information and hence on the analysis phase or decisions taken. An incorrect missing value handling, in fact, can impact negatively the effectiveness of the other tasks as the forecasting [4] or anomaly detection and, in the Smart Grid context, can

lead to severe failures as the mismatch between supply and demand for an incorrect scheduling, or to a voltage violation a wrong regulation may lead to [5]. In the presence of multiple time-series with different characteristics and hence possible issues, such as in Smart Grid context, the problem of managing the data in an aggregated way emerge naturally and it is at the base of the data fusion techniques. For this reason we need a solution that fuses information coming from different sources and that is robust to missing data leveraging the intra- and inter- correlation present in the different observed data. In this way we can have a robust method to missing data for the next downstream tasks and, using more information, can improve the decision making capability.

Regarding the missing data handling, a best practice used in the power industry is to apply linear interpolation for missing or corrupted values in intervals shorter than two hours. For intervals longer than two hours, a typical profile is used considering historical data and taking into account the day in the week and the presence of holidays [6]. Many other approaches to impute missing data in time series have been proposed over the years. Some of them leverage the similarity of neighbors [7], others use Expectation-Maximization (EM) methods, autoregressive-moving average models (ARIMA),

✉ Amedeo Buonanno
amedeo.buonanno@enea.it

Extended author information available on the last page of the article

or Kalman filter models [8]. More recently, Deep Learning approaches have been applied to the problem of missing data imputation in time series [9] such as the Recurrent Neural Networks [10] and the Generative Adversarial Networks [11].

The handling of missing data in the fusion process has been investigated in several works [12, 13]. The most used approaches are based on classical data imputation, such as interpolation, averages, etc., applied to the time series before data combination.

In this work we improve the classical methods using a machine learning approach based on data history. We propose a DaI-FeO/DaI-DaO Fusion architecture (based on the Dasarathy classification [14]), where the data inputs, i.e. the power consumption measured by different sub-meters, are fused to build a representation robust to missing values (feature output) in order to improve the downstream tasks, and to provide better imputation (data output).

In this work we focus on the Autoencoder-based model because it is a very powerful and flexible approach to data fusion and, at the same time, it can be designed and trained to be robust to missing data. In this way it is possible to represent the observed data in a lower space that can be used for downstream tasks such as clustering, classification, forecasting, etc.

Fusion models based on Autoencoders have been investigated in different works [15–18], also in the energy context [19–22]. The majority of them uses the Autoencoder as a feature extractor inserted in a more complex architecture to perform downstream supervised tasks.

In the presence of missing or corrupted data it is important to learn a data representation which is robust to the contamination. This can be achieved using the Denoising Autoencoder [23]. When trained using a particular masking noise, the model can learn to fill-in missing values thanks to dependencies present in the input data.

The process of denoising has also an interesting geometric interpretation based on the *manifold assumption* (the natural high-dimensional data lay on a non-linear low-dimensional manifold). Using the denoising criterion during training, we learn to map corrupted examples (likely farther from embedding manifold) to the uncorrupted version of them (on the manifold). If the corrupted example is very far from the manifold, the Denoising Autoencoder has to make a considerable effort to generate a correct value [24]. The usage of Autoencoder as model to replace missing values has been already suggested in [23] but, more recently, other works have used a similar approach [25].

In [17] the authors have investigated a data fusion architecture for missing data imputation where the reconstruction task can benefit from having other signals available. Contextual information, in fact, have demonstrated to be very effective in various data processing applications, such as

scene understanding [26] or language models [27], and it has been exploited in our previous work [28] where we have investigated a Bayesian approach for FeI-DeO Fusion based on the Factor Graph paradigm [29–31], showing how to effectively manage missing, or wrong values, also taking into account sensor reliability in selected sensors, or in selected measurements.

In this work, we propose a new architecture based on Autoencoder model, convolutional layers, skip connections and ad-hoc augmented training sets, to impute missing data using a shared embedding space, that fuses information coming from different sensors. A similar approach has also been investigated in [17], but in our work, more signals to fuse are considered, the convolutional architecture is employed for the specific autoencoders and for the fusion layer, and an ad-hoc augmented training set is considered.

The main contribution of this work is to show that the imputation of missing data, in the energy context, in presence of multiple sensors, in very challenging but real situation where large portions of the signals are missing, can be helped by a proper data augmentation scheme and by the information carried out by other signals properly combined. Moreover, the use of an Autoencoder approach allows to obtain a compressed representation of the input signals which is robust to the missing data and that can be used for other downstream tasks such as clustering, forecasting, etc. To our knowledge no previous work has used this type of architecture and performed this type of analysis in the energy context.

The description of the architecture used for the fusion of the sensor signals is presented in Section 2. The different missing data patterns and the several types of augmentation are presented in Section 3 and Section 4, respectively. The datasets and evaluation method used in our work are presented in Section 5 and Section 6, respectively. The obtained results and discussion of experiments are presented in Section 7. Finally, in Section 8, conclusions and suggestions for further work are presented.

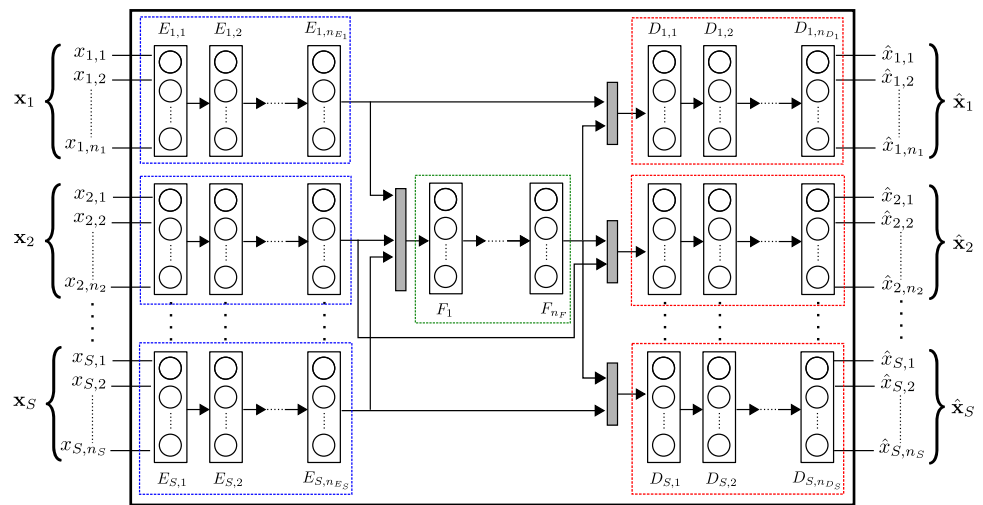
2 Sensor Fusion Model

The proposed feature fusion architecture is based on the concept of sharing the intra- and inter-modal correlation of several signals involved [17].

As depicted in Fig. 1, we have S different sensors, the generic i -th sensor produces n_i measurements (e.g. related to n_i contiguous timestamps) globally denoted as signal $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n_i}]$. The dataset is composed by N records: $\{\mathbf{x}_i^{(n)}\}_{i=1:S}^{n=1:N}$.

For each record of the i -th sensor, the n_i measurements feed n_{E_i} specific Encoding Layers $\mathbf{E}_i = \{E_{i,1}, E_{i,2}, \dots, E_{i,n_{E_i}}\}$ (blue boxes in Fig. 1).

Fig. 1 Conceptual scheme of Data Fusion Model based on Autoencoder



The details of the encoding process for the generic i -th signal are described in (1) where: $\beta_{i,j}$ is the bias for the j -th Encoding Layer of the i -th signal and σ_e is the activation function:

$$\begin{aligned} \mathbf{e}_{i,1} &= \sigma_e(E_{i,1} \cdot \mathbf{x}_i + \beta_{i,1}) \\ \mathbf{e}_{i,2} &= \sigma_e(E_{i,2} \cdot \mathbf{e}_{i,1} + \beta_{i,2}) \\ &\vdots \\ \mathbf{e}_{i,n_{E_i}} &= \sigma_e(E_{i,n_{E_i}} \cdot \mathbf{e}_{i,n_{E_i}-1} + \beta_{i,n_{E_i}}) \end{aligned} \tag{1}$$

All the encoded signals, $\{\mathbf{e}_{i,n_{E_i}}\}_{i=1}^S$ are concatenated as described in (2) where $[\cdot, \cdot]$ is the concatenation operator:

$$\mathbf{f}_0 = [\mathbf{e}_{1,n_{E_1}}, \mathbf{e}_{2,n_{E_2}}, \dots, \mathbf{e}_{S,n_{E_S}}] \tag{2}$$

Then \mathbf{f}_0 feeds n_F shared Encoding Layers $\mathbf{F}_i = \{F_1, F_2, \dots, F_{n_F}\}$ (green box in Fig. 1). The process of the fused encoding is described in (3) where $\beta_{f,j}$ is the bias for the j -th shared Encoding Layer and σ_f is the activation function:

$$\begin{aligned} \mathbf{f}_1 &= \sigma_f(F_1 \cdot \mathbf{f}_0 + \beta_{f,1}) \\ \mathbf{f}_2 &= \sigma_f(F_2 \cdot \mathbf{f}_1 + \beta_{f,2}) \\ &\vdots \\ \mathbf{f}_{n_F} &= \sigma_f(F_{n_F} \cdot \mathbf{f}_{n_F-1} + \beta_{f,n_F}) \end{aligned} \tag{3}$$

Finally, specific n_{D_i} Decoding Layers $\mathbf{D}_i = \{D_{i,1}, D_{i,2}, \dots, D_{i,n_{D_i}}\}$ (red boxes in Fig. 1), reconstruct the n_i measurements of the generic i -th sensor: $\hat{\mathbf{x}}_i = [\hat{x}_{i,1}, \hat{x}_{i,2}, \dots, \hat{x}_{i,n_i}]$. In detail, the encoded signal $\mathbf{e}_{i,n_{E_i}}$ for each sensor is concatenated to the fused encoded signal \mathbf{f}_{n_F} as described in (4):

$$\mathbf{d}_{i,0} = [\mathbf{e}_{i,n_{E_i}}, \mathbf{f}_{n_F}] \tag{4}$$

The process of the decoding is described in (5) where $\zeta_{i,j}$ is the bias for the j -th Decoding Layer of the i -th signal and σ_d is the activation function:

$$\begin{aligned} \mathbf{d}_{i,1} &= \sigma_d(D_{i,1} \cdot \mathbf{d}_{i,0} + \zeta_{i,1}) \\ \mathbf{d}_{i,2} &= \sigma_d(D_{i,2} \cdot \mathbf{d}_{i,1} + \zeta_{i,2}) \\ &\vdots \\ \mathbf{d}_{i,n_{D_i}} &= \sigma_d(D_{i,n_{D_i}} \cdot \mathbf{d}_{i,n_{D_i}-1} + \zeta_{i,n_{D_i}}) \end{aligned} \tag{5}$$

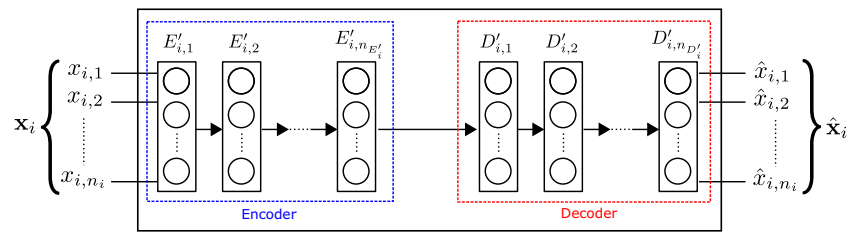
Both \mathbf{E}_i and \mathbf{F}_i can be Dense Layers, or Convolutional Layers, while the various \mathbf{D}_i can be Dense Layers, or Transpose Convolutional Layers. Skip connections (using concatenation as in DenseNet [32]), are introduced to accelerate the learning process, and for each signal, the last Decoding Layer $D_{i,n_{D_i}}$, is a dense layer with dimension n_i .

The loss function is computed as weighted sum of the MSE for each sensor between the reconstructed signal and the ground truth. Since each sensor can suffer of specific problems, and be corrupted following a particular pattern, we define the indexes of the corrupted signal components as $\mathcal{S}_i = \{j : j \in \{1, \dots, n_i\}, x_{i,j} \text{ is corrupted}\}$. Hence, the loss function becomes:

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^S w_i \cdot \left(\alpha \cdot \frac{1}{|\mathcal{S}_i|} \left(\sum_{j \in \mathcal{S}_i} (x_{i,j} - \hat{x}_{i,j})^2 \right) \right. \\ &\quad \left. + \beta \cdot \frac{1}{n_i - |\mathcal{S}_i|} \left(\sum_{j \notin \mathcal{S}_i} (x_{i,j} - \hat{x}_{i,j})^2 \right) \right) \end{aligned} \tag{6}$$

where w_i is the weight related to each sensor, and α and β are the weights for the reconstruction error on components that are, respectively, corrupted or not. If the weights w_i are equal for all sensors, and α and β are chosen properly (e.g., $\alpha = |\mathcal{S}_i|$ and $\beta = n_i - |\mathcal{S}_i|$), we obtain a value that is propor-

Fig. 2 Conceptual scheme of the specialized Autoencoder for each sensor. The blue and red boxes contain, respectively, the encoder and the decoder parts



tional to the global MSE computed over all measurements of all sensors.

For each sensor, there is a specialized Autoencoder, as depicted in Fig. 2, composed by an Encoder and a Decoder (respectively blue and red box in the Fig. 2). At the end of the training process of each Autoencoder, the learned weights for the Encoding Layers $\{E'_i\}_{i=1}^{n_{E'_i}}$ are used as weights (or could be used as initial weights in case of fine tuning) of the Encoding Layers of the overall architecture (Fig. 1 and relative blue boxes). The learned weights for the Decoding Layers $\{D'_i\}_{i=1}^{n_{D'_i}}$ are instead discarded. In this work we have used a symmetric Autoencoder ($n_{D'_i} = n_{E'_i} - 1$) and the same number of neurons, or filters, used for the Encoding Layers have been used for the Decoding Layers in the inverse order. When the specialized Autoencoder is convolutional, the last layer is a CONV 1D 1x1.

3 Missing data patterns

Figure 3 shows an example with three sensors with six measurements each. The first row (Fig. 3(a)) has no missing data. The other rows show typical missing patterns that can occur in real data for a single sensor: randomly distributed (Fig. 3(b)), in the central part (Fig. 3(c)), in the last part (Fig. 3(d)), in the initial part (Fig. 3(e)).

The presence of missing data in real context is critical as reported in [22] where in a real smart meter dataset of 50

million of load measurements, there are totally 420k missing points (1% of total), 34k isolated missing points and 38k missing contiguous blocks. Usually, random missing patterns (Fig. 3(b)) are observed when there are communications, or sensor issues, of brief duration (intermittent failure). Contiguous values may be missing when a sensor, or its connection, stops working for a while before reconnection (prolonged failure) [5].

These patterns should be seen as typical day data where the central hours may correspond to peaks of energy demand, or photovoltaic energy production. Having the central part of these time series completely missing is, hence, one of the worst situations to tackle.

To solve these problems, straightforward imputation methods have been suggested, such as replace the missing values with template values obtained from the training set, or leveraging some statistics of the signal estimated on historical data (e.g. average, median, etc.) [6], or using algorithms such as the popular MICE [33]. In our approach, instead, the imputation is relying completely on the trained Autoencoder that fills-in automatically the missing parts.

4 Augmented training set

In solving real problems using machine learning, the training algorithms need to have available rich data sets that contain with sufficient frequency the patterns of interest. When this

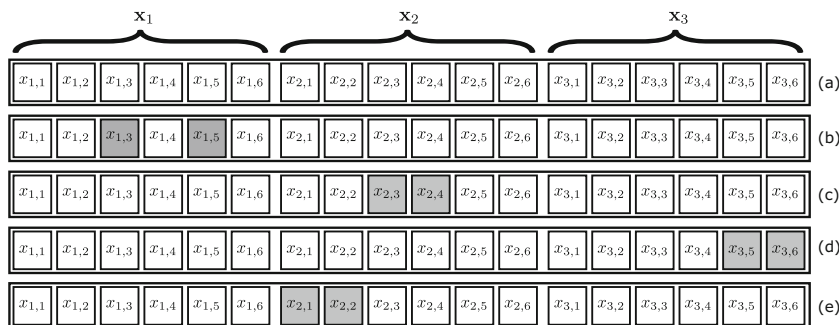
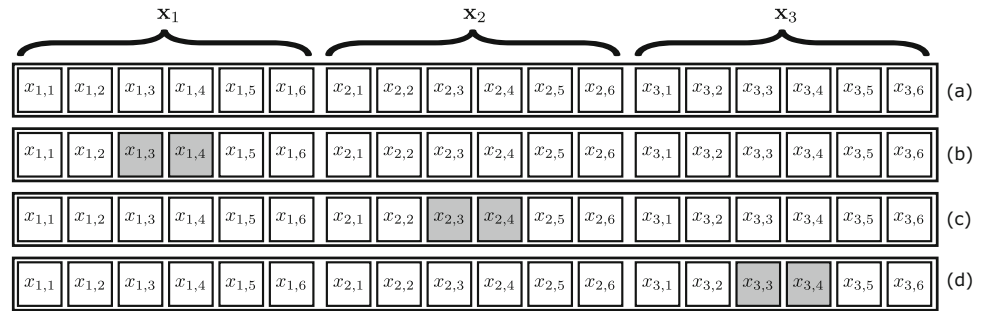


Fig. 3 Missing Data Patterns for a configuration composed by three sensors with six measurements each. (a) No missing Data; (b) 33% of missing data randomly distributed on measurements of first sensor; (c) 33% of missing data on the central part of measurements of second

sensor; (d) 33% of missing data on the last part of measurements of third sensor; (e) 33% of missing data on the first part of measurements of second sensor

Fig. 4 Augmented training set for a configuration composed by three sensors with six measurements each. (a) Original Record; New record obtained removing the central 33% of the measurements for: (b) the first signal; (c) the second signal; (d) the third signal



is not possible, it is often necessary to use *data augmentation*, i.e. enrich the training set creating artificially the critical situations to be addressed.

Following the discussion in Section 3, we focus on the missing data in the central part. To reduce the dataset shift [34], between the training data and the effective situations that can occur, we have created an augmented training set as depicted in Fig. 4: for each original record, we create S new (synthetic) records containing, for each signal, the central part completely removed keeping the rest. Then the original record is used as the desired output (ground truth).

In the following sections we present the results using the network architecture depicted in Fig. 1, with the main hyperparameters listed in Table 1, and trained using different types of augmentation. Each type of augmentation defines a particular model to test:

- AE: original training set (the records as depicted in Fig. 4(a))
- AE-A: augmented training set (the records as depicted in Fig. 4(a), (b), (c), (d))
- AE-A-ALL: augmented training set (the records as depicted in Fig. 4(a), (b), (c), (d)) and adding new records with missing data randomly distributed for each signal (K_{all} repetitions)
- AE-A-ONLY-SYNTH: training set composed only by the synthetic records (the records as depicted in Fig. 4(b), (c), (d))
- AE-A-CONTIG: training set composed only by the synthetic records obtained removing contiguous samples for each signal with center position randomly distributed (K_{contig} repetitions)

Other two models have been tested with main hyperparameters listed in Table 2:

- AE-A-ALL-CNN: augmented training set as for the model AE-A-ALL
- AE-A-ONLY-SYNTH-CNN: training set composed as for the model AE-A-ONLY-SYNTH

The following architectures have been tested as reference models:

- AE-S: the model based on Stacked Sparse Autoencoder, trained using the original training set (the records as depicted in Fig. 4(a)) and with the main hyperparameters listed in Table 3. This model is similar to model described in [17] and some choices have been made to make it comparable with other proposed architectures (e.g., not use of the layer-wise pretraining, absence of fine tuning procedure)
- AE-D: the model based on Denoising Autoencoder [23] trained using the original training set (the records as depicted in Fig. 4(a)) and using the Dropout on input layer
- IMPUTER: the Multiple Iterative Imputer (MICE) [33]
- BASELINE: the baseline that substitutes the missing values with average of the signal computed on training set for that time step

5 Dataset

The datasets we have used for experiments are: REFIT [35] and the "Individual household electric power consumption Data Set" in the UCI Machine Learning Repository [36].

Table 1 Configuration for the tested architectures with dense shared encoder

Hyperparameter	Values
E_i	Conv 1D [50, 25] - Kernel Size [1x3] - selu
F_i	Dense [50] - sigmoid
D_i	Conv 1D Transpose [25, 50] - selu
Learning Rate	0.01, Reduced on Plateau
Batch Size	32
Optimizer	SGD with momentum 0.9 (Nesterov)
Max. Num. Epoch	1000
Notes	Early Stopping with Patience 10 steps and Refit

Table 2 Configuration for the tested architectures with convolutional shared encoder

Hyperparameter	Values
E_i	Conv 1D [50, 25] - Kernel Size [1x3] - selu
F_i	Conv 1D [10] - sigmoid
D_i	Conv 1D Transpose [25, 50] - selu
Learning Rate	1e-5, Reduced on Plateau
Batch Size	32
Optimizer	Adam
Max. Num. Epoch	1000
Notes	Early Stopping with Patience 10 steps and Refit

The first dataset includes cleaned electrical consumption data in Watts for 20 households at aggregate and appliance level, sampled each 8 second for the period from October 2013 to June 2015. We have focused on house 15 and the following appliances: Appliance2 (tumble dryer), Appliance3 (dishwasher), Appliance5 (computer site) and Appliance6 (television site).

The second dataset contains minute-wise power consumption measurements gathered from a house located in France between December 2006 and November 2010 (47 months) with 3 sub_meters:

- sub_metering_1 related to the kitchen, containing mainly a dishwasher, an oven, and a microwave;
- sub_metering_2 related to the laundry room, containing a washing-machine, a tumble-dryer, a refrigerator, and a light;
- sub_metering_3 connected to electric water-heater and an air-conditioner.

The consumption of the house related to other rooms/appliances has been taken in account as difference between the global active power and the active power measured by three sub_meters, named sub_metering_4.

For both datasets we have resampled the original time series on hour basis using the average as aggregation method. Hence we have 4 sensors and each one contains 24 mea-

surements for each day. From the complete dataset we have considered, for each month, 75% of data for training and the 25% for test. In this way the training set and test set contain information coming from all available months.

6 Model evaluation

For the i -th sensor, each signal x_i has been normalized using standardization, that is subtracting the mean and dividing by the standard deviation computed on all signals x_i in the training set.

After the training phase, the models have been tested to reconstruct input signals belonging to the test set with and without missing values. The desired behavior is the following:

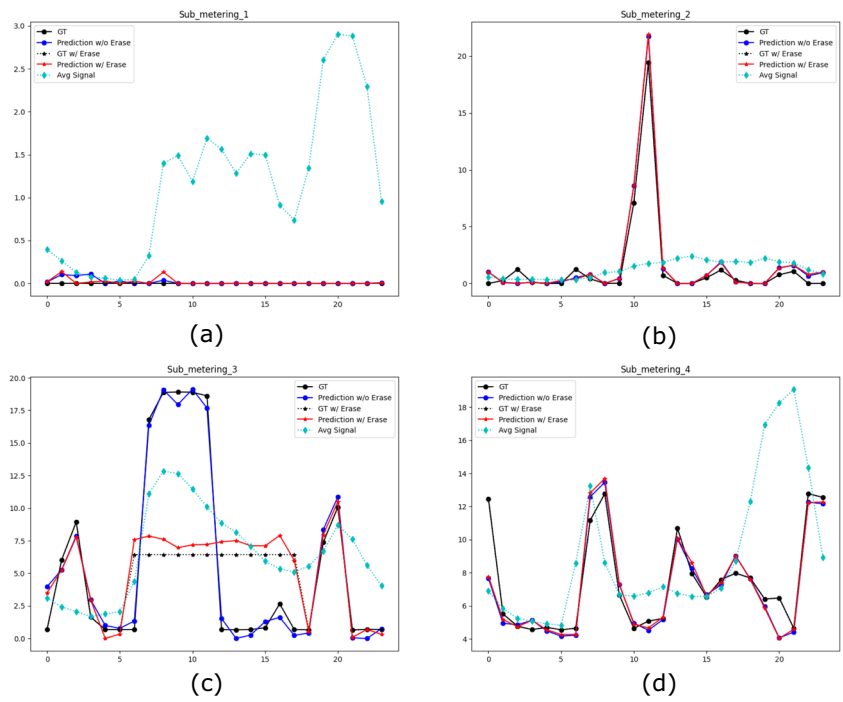
- For the input signals without missing values the model should reconstruct the signal at best even though it has not been seen during the training phase.
- For the input signals with missing values distributed using the patterns depicted in Fig. 3, the model should try to impute the missing values obtaining results as close as possible to the ground truth signal.

To simulate the situations that can happen in the real contexts, we consider the erasures with random patterns (Fig. 3(b)) and

Table 3 Configuration for the model based on Stacked Sparse Autoencoder

Hyperparameter	Values
E_i	Dense [50, 25] - selu
F_i	Dense [50] - sigmoid
D_i	Dense [25, 50] - selu
Learning Rate	0.01, Reduced on Plateau
Batch Size	32
Optimizer	SGD with momentum 0.9 (Nesterov), Sparse Weight: 0.1, Sparsity Parameter: 0.1
Max. Num. Epoch	1000
Notes	Early Stopping with Patience 10 steps and Refit

Fig. 5 Reconstruction results on UCI Test set for 4 signals for model AE. Only signal sub_metering_3 contains erasures on central part. Reconstruction for signal: (a) sub_metering_1; (b) sub_metering_2; (c) sub_metering_3; (d) sub_metering_4



central patterns (Fig. 3(c)). The other two patterns (Fig. 3(d), (e)) are similar and don't add anything to our discussion. During the following experiments the missing values have been set to a fixed value that usually is zero (average value when the signal is denormalized).

These experiments have been performed with the complete information coming from other sensors, or with their complete absence. In this way we can observe the importance of the fused representation in the shared Encoding Layers for imputing tasks.

Fig. 6 Reconstruction results on UCI Test set for 4 signals for model AE-A. Only signal sub_metering_3 contains erasures on central part. Reconstruction for signal: (a) sub_metering_1; (b) sub_metering_2; (c) sub_metering_3; (d) sub_metering_4

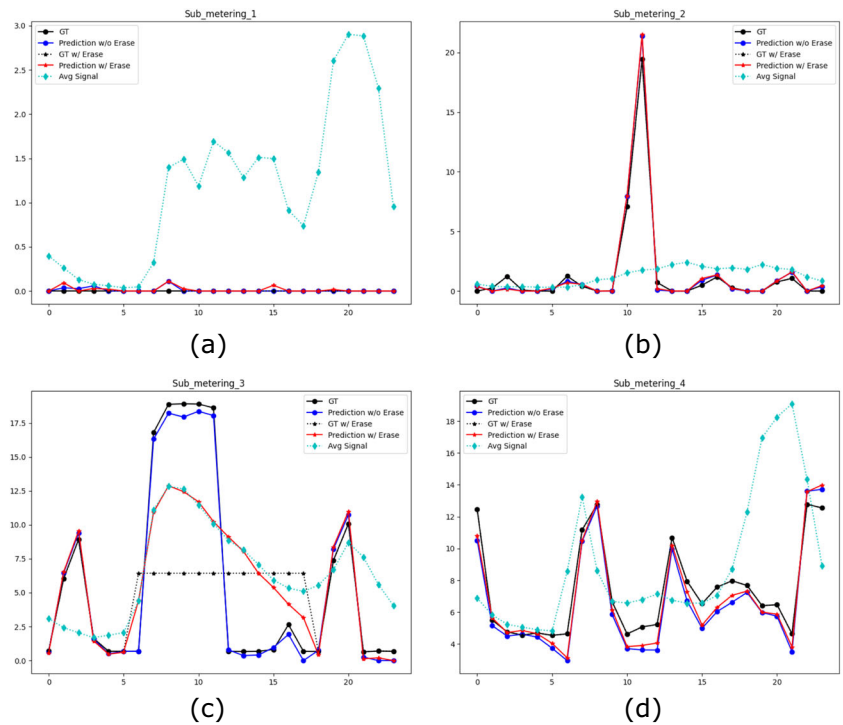
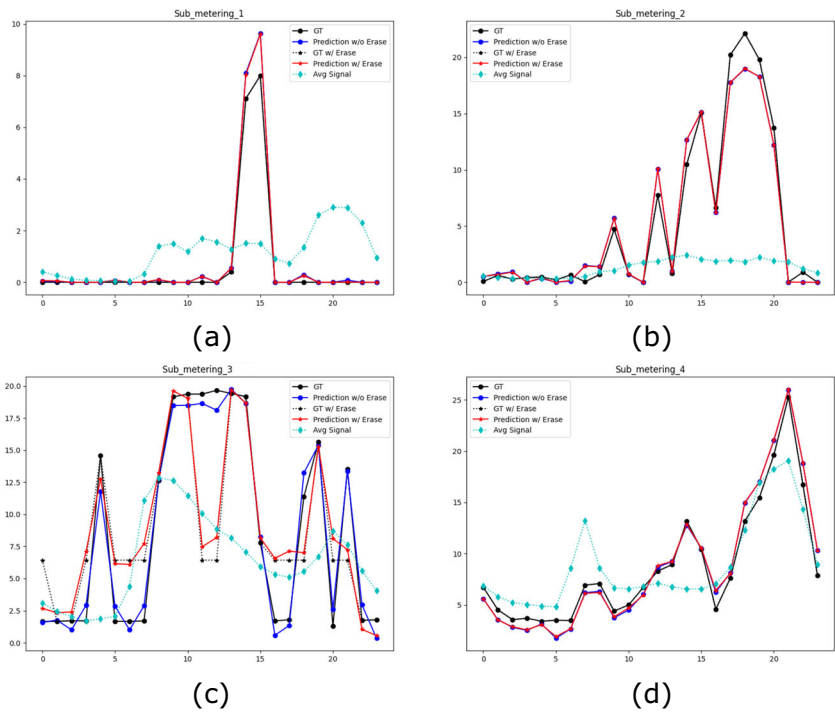


Fig. 7 Reconstruction results on UCI Test set for 4 signals for model AE. Only signal sub_metering_3 contains random erasures. Reconstruction for signal: (a) sub_metering_1; (b) sub_metering_2; (c) sub_metering_3; (d) sub_metering_4



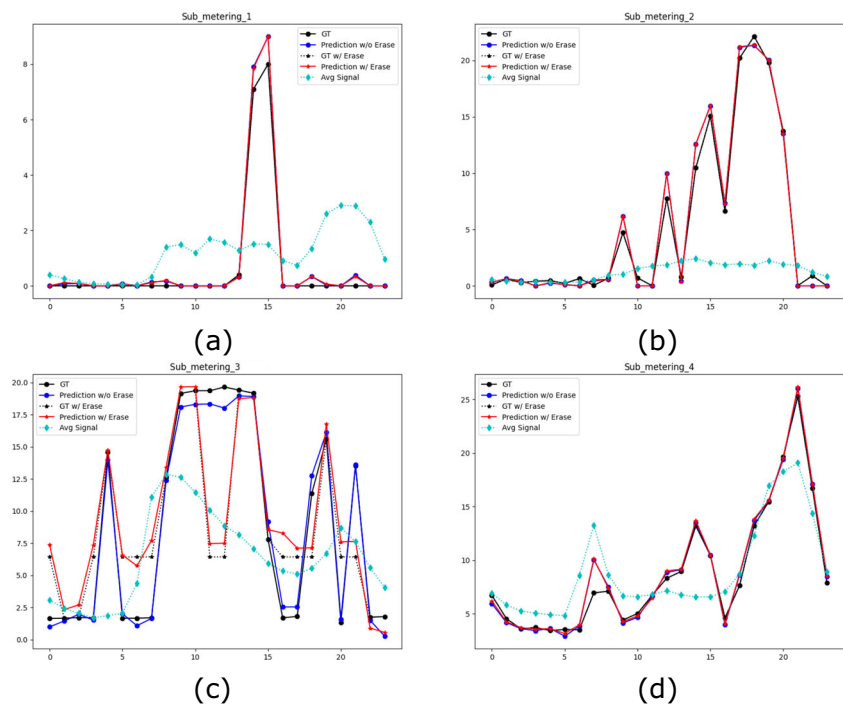
7 Results and discussion

The simulations have been performed considering the weights $w_i = 1$ for all sensors, the weight for corrupted points in the loss computation $\alpha = 0.7$ and $\beta = 1 - \alpha$. The number of repetitions in the data augmentation are: $K_{all} = 3$, $K_{contig} = 10$. The percentage of the missing data for augmented dataset

is 50% and the same percentage is used for dropout rate in AE-D.

In the following figures we show the prediction results of AE and AE-A models for UCI dataset, with three signals with no erasures (sub_metering_1, sub_metering_2 and sub_metering_4), and sub_metering_3 signal with half signal completely removed:

Fig. 8 Reconstruction results on UCI Test set for 4 signals for model AE-A. Only signal sub_metering_3 contains random erasures. Reconstruction for signal: (a) sub_metering_1; (b) sub_metering_2; (c) sub_metering_3; (d) sub_metering_4



- In the central part (Figs. 5 and 6).
- In random positions (Figs. 7 and 8).

In Figs. 5, 6, 7, 8, and 9: the black solid line is the ground truth; the blue line is the result of the prediction using the ground truth as input; the black dashed line is the input containing the erasures (if any); the red line is the result of the prediction using the input containing the erasures; the cyan dashed line is the average signal computed over all training set.

The AE-A model has been trained using a training set augmented using 50% of erased samples in the central zone, hence the training set for AE-A model has been augmented considering also signals with the 12 central hours completely erased.

In the Fig. 6(c) we show that using the AE-A model, with central part of sub_metering_3 input signal completely removed, the reconstructed signal (red line) doesn't follow the black dashed line (representing the signal with erasures), but tries to follow the black solid line that represents the original signal without erasures (not provided as input to the model). This means that the model is able to partially impute the missing values and not simply replicates the inputs that contains the missing values as the AE model does (Fig. 5(c)). The same behavior is observed also in AE-A-ALL, AE-A-ONLY-SYNTH, AE-A-ALL-CNN and AE-A-ONLY-SYNTH-CNN models, not shown here for brevity.

The AE-A model leverages the behavior observed in the central area during the training phase and hence the estimation of the average of the corrupted sub_metering_3 signal (cyan line in Fig. 6(c)). Moreover, the erasure of

sub_metering_3 doesn't impact too negatively on the reconstruction of the other signals (red lines and blue lines in Fig. 6(a), (b), (d)).

In the Figs. 7(c) and 8(c) we show, respectively, the results of AE and AE-A models when 50% of samples, randomly distributed, of sub_metering_3 has been removed. For AE model is confirmed an absence of robustness to the missing values, but now also the reconstruction results of AE-A are not so good. The reason is that the erasures can occur in configuration that the model didn't see during the training process (AE-A has been trained using dataset containing signals with only the central part deleted).

If we consider the result using AE-D model (Fig. 9(c)), we can note that the model is more robust to the erasures as it is filtering them out, but other signals are reconstructed with more difficulty.

This behavior is confirmed also if we evaluate the reconstruction error on all test set as described in Tables 4 and 5, where there are listed the RMSE values for models AE, AE-A, AE-A-ONLY-SYNTH, AE-A-ALL, AE-CONTIG, AE-A-ONLY-SYNTH-CNN, AE-A-ALL-CNN, AE-S, AE-D, IMPUTER and BASELINE applied to, respectively, UCI and REFIT datasets. The evaluation is performed on test set records with the 50% of the central part of *i*-th signal completely removed. The RMSE between the reconstructed and ground truth signals (without erasures) of the test set is computed only considering the samples that have been erased.

In particular, the AE model is confirmed not being good solution for obtaining a robust representation of the input data with missing data in the central part.

Fig. 9 Reconstruction results on UCI Test set for 4 signals for model AE-D. Only signal sub_metering_3 contains random erasures. Reconstruction for signal: (a) sub_metering_1; (b) sub_metering_2; (c) sub_metering_3; (d) sub_metering_4

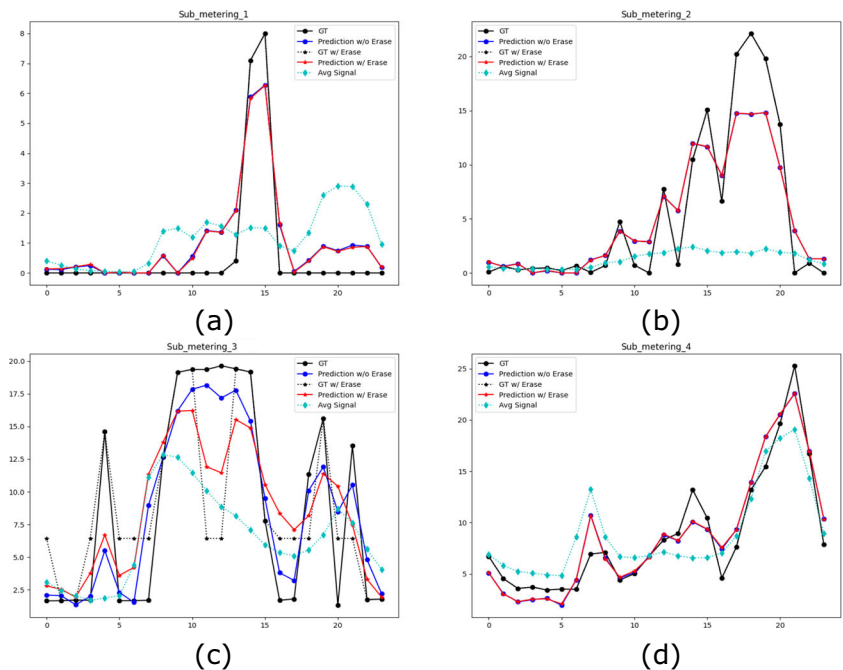


Table 4 Reconstruction error only on erased samples for UCI dataset on Test set for the considered models. One half of the input signal, in the central part, has been erased before it has been presented to the network. Column S_i stands for Sub_metering_i

	S ₁	S ₂	S ₃	S ₄
AE	2.741	3.447	7.807	5.250
AE-A	2.411	3.208	7.113	3.750
AE-A-ONLY-SYNTH	2.281	3.194	7.077	3.628
AE-A-ALL	2.350	3.165	7.164	3.997
AE-A-CONTIG	2.360	3.234	6.903	3.813
AE-A-ONLY-SYNTH-CNN	2.324	3.067	6.300	3.082
AE-A-ALL-CNN	2.430	3.108	6.125	3.465
AE-S	2.812	3.549	7.560	5.490
AE-D	2.517	3.410	7.438	4.537
IMPUTER	2.359	3.257	6.807	3.525
BASELINE	2.448	3.358	7.332	4.597

For UCI dataset, the models with shared convolutional layers are the best choices. AE-A-ONLY-SYNTH-CNN outperforms other models (except for Sub_metering₁ and Sub_metering₃ where it is the second best model after AE-A-ONLY-SYNTH and AE-A-ALL-CNN model, respectively) and AE-A-ALL-CNN outperforms other models for Sub_metering₃ and it is the second best model for Sub_metering₂ and Sub_metering₄ after AE-A-ONLY-SYNTH-CNN. Moreover, AE-A-ALL-CNN model presents an average improvement over IMPUTER of about 3%, over AE of about 19% and over AE-D of about 13%.

Also for REFIT dataset, the models with shared convolutional layers are the best choices. AE-A-ALL-CNN outperforms other models (except for Appliance₂) and AE-A-ONLY-SYNTH-CNN is in the top 5 models (except for

Table 5 Reconstruction error only on erased samples for REFIT dataset on Test set for the considered models. One half of the input signal, in the central part, has been erased before it has been presented to the network. Column A_i stands for Appliance_i

	A ₂	A ₃	A ₅	A ₆
AE	56.169	73.840	3.319	22.871
AE-A	47.640	73.533	2.292	20.586
AE-A-ONLY-SYNTH	42.277	73.398	2.252	20.529
AE-A-ALL	44.270	69.980	2.532	20.575
AE-A-CONTIG	43.009	70.253	2.251	20.985
AE-A-ONLY-SYNTH-CNN	43.478	75.344	2.207	19.769
AE-A-ALL-CNN	45.913	66.988	1.750	19.751
AE-S	41.980	72.206	2.967	21.549
AE-D	50.212	75.212	3.144	22.137
IMPUTER	49.851	70.048	2.549	20.771
BASELINE	41.110	71.033	2.748	21.399

Table 6 Reconstruction error only on erased samples for UCI dataset on Test set for the considered models. One half of the input signal, in the random position, has been erased before it has been presented to the network. Column S_i stands for Sub_metering_i

	S ₁	S ₂	S ₃	S ₄
AE	2.811	2.849	6.643	6.060
AE-A	2.787	2.745	7.038	6.506
AE-A-ONLY-SYNTH	2.720	2.734	7.007	6.451
AE-A-ALL	2.509	2.618	5.854	5.116
AE-A-CONTIG	2.641	2.707	6.517	5.843
AE-A-ONLY-SYNTH-CNN	2.758	2.747	6.818	6.399
AE-A-ALL-CNN	2.375	2.502	5.250	4.505
AE-S	2.954	3.049	6.761	6.552
AE-D	2.488	2.597	5.625	4.743
IMPUTER	2.351	2.440	4.998	3.906
BASELINE	2.682	2.788	6.400	5.977

Appliance₃). AE-A-ALL-CNN model shows an average improvement over IMPUTER of about 12%, over AE of about 22% and over AE-D of about 19%. These results show how a proper augmentation dataset, with missing values distributed as expected in a real situation (in this case in the central part), together with a convolutional fusion layer, is a preferable choice respect to other presented solutions. In particular, if we focus only on the two most employed approaches (AE-D and IMPUTER), in presence of one half of the input signal, in the central part, completely erased, AE-A-ALL-CNN presents an improvement of the imputation capability in the average of about 12%.

If the erasure pattern changes, for instance, the samples are removed randomly (Tables 6 and 7), we can observe an interesting behavior. For UCI dataset, AE-A-ALL-CNN,

Table 7 Reconstruction error only on erased samples for REFIT dataset on Test set for the considered models. One half of the input signal, in the random position, has been erased before it has been presented to the network. Column A_i stands for Appliance_i

	A ₂	A ₃	A ₅	A ₆
AE	34.676	65.353	2.519	20.120
AE-A	34.487	70.744	2.563	23.368
AE-A-ONLY-SYNTH	30.393	69.467	2.516	22.795
AE-A-ALL	30.818	64.690	2.241	19.112
AE-A-CONTIG	30.588	66.240	2.448	22.241
AE-A-ONLY-SYNTH-CNN	34.892	72.457	3.036	26.000
AE-A-ALL-CNN	34.039	55.018	1.651	15.226
AE-S	33.223	68.727	2.855	21.189
AE-D	31.526	65.174	2.340	17.793
IMPUTER	39.145	62.670	1.699	15.467
BASELINE	27.686	64.593	2.551	19.993

is the second best model after IMPUTER, while in REFIT dataset, it is able to outperform other approaches (except for Appliance_2). Moreover, AE-A-ALL-CNN model, for UCI dataset, presents an average improvement over UAE of about 19% and over UAE-D of about 5%, and for REFIT dataset, it presents an average improvement over IMPUTER of about 7%, over UAE of about 19% and over UAE-D of about 13%.

These simulations seem to confirm that training the model modifying the input patterns following the same percentage and missing pattern that can be observed in the real context, helps the imputation of missing values. The prior information on the missing data process can help to construct a more robust representation. Often this information is not available and hence we can augment the dataset adding records that contains erasures distributed following several patterns that, eventually, can occur as with AE-A-ALL or AE-A-ALL-CNN models. The latter emerges as a very good model in both situations, when the missing patterns are distributed in the central part or randomly.

These results suggest that, designing properly the data augmentation phase, we can make the representation more robust to some particular missing patterns than others. In this way we can, for example, impute missing data that follow a

particular missing pattern, but be transparent to other types of corruption that it is necessary to be “transmitted” to following task in the pipeline (e.g. anomaly detector).

In order to assess the impact of the augmentation and of the convolutional neural network as shared encoder, in the Fig. 10 we show the imputation results with the classical Autoencoder approach, then we introduce the data augmentation and finally we employ the convolutional fusion layer. From the graph it is evident how the proposed solutions improve the imputation capability of the architecture and that both convolutional shared encoder and the augmentation have an important role in the final results.

The architecture is built in order to take advantage also from other available signals. In the following we assess the role of the other signals (not corrupted) in the reconstruction capability of the proposed architecture. In the Figs. 12 and 11 we show how RMSE values for AE, AE-A, AE-A-ONLY-SYNTH, AE-ALL, AE-CONTIG, AE-A-ONLY-SYNTH-CNN, AE-A-ALL-CNN, AE-D vary when the *i*-th signal contains 50% of samples completely removed in the central part and other signals are progressively completely removed. These figures confirm how the usage of other signals can improve the imputation performance of the considered models even though some signals are more sensitive than others, probably because the correlation among the signal and other ones is not so high. Usually, the lowest value of RMSE for all fours signals is obtained when other three signals are available (0 signal erased) and the model can leverage the information of other signals. Removing information coming from other signals, the reconstruction error on the *i*-th signal increases, confirming the importance of the data fusion in the imputation of missing values.

8 Conclusion

The necessity of a robust method to fill-in missing data is important in IoT context and in particular when we use information coming from several sensors for making decision as in the Smart Energy Grid. In this work we have proposed an Autoencoder-based data fusion architecture that can achieve these objectives. The model is completely data-driven and leverages information coming from other sensors in order to improve the imputation performance. Our technique has been tested in very challenging situations where the most important part of the signals may be completely lost. We have shown that a dedicated data augmentation phase is a crucial step in making the Autoencoder representation robust to missing patterns. Specific augmentation patterns could be effectively used for making this paradigm very versatile. The proposed approach, as any data-driven solution, is dependent on training data, their quality and quantity. If the training data is biased, not representative of the population or of the miss-

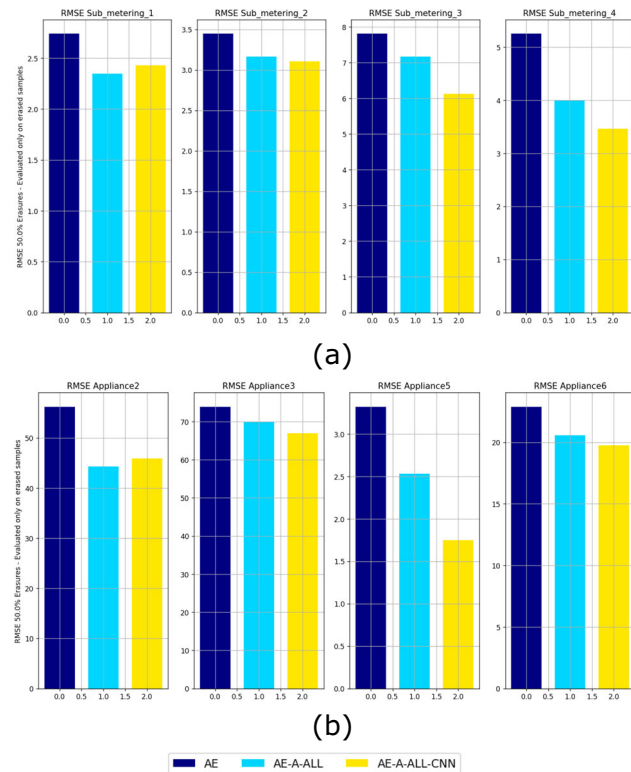


Fig. 10 Reconstruction error only on central erased samples for (a) UCI dataset and (b) REFIT dataset without data augmentation (AE), with data augmentation and the dense shared Encoder (AE-ALL), and with data augmentation and convolutional shared Encoder (AE-ALL-CNN)

Fig. 11 Impact of other signals to the reconstruction of the signal on UCI dataset

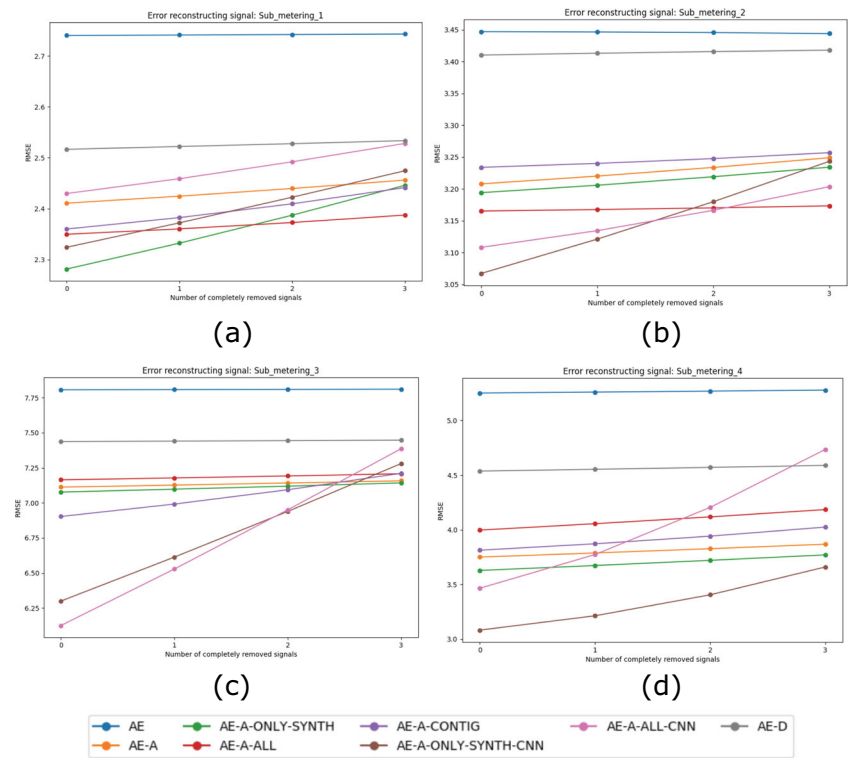
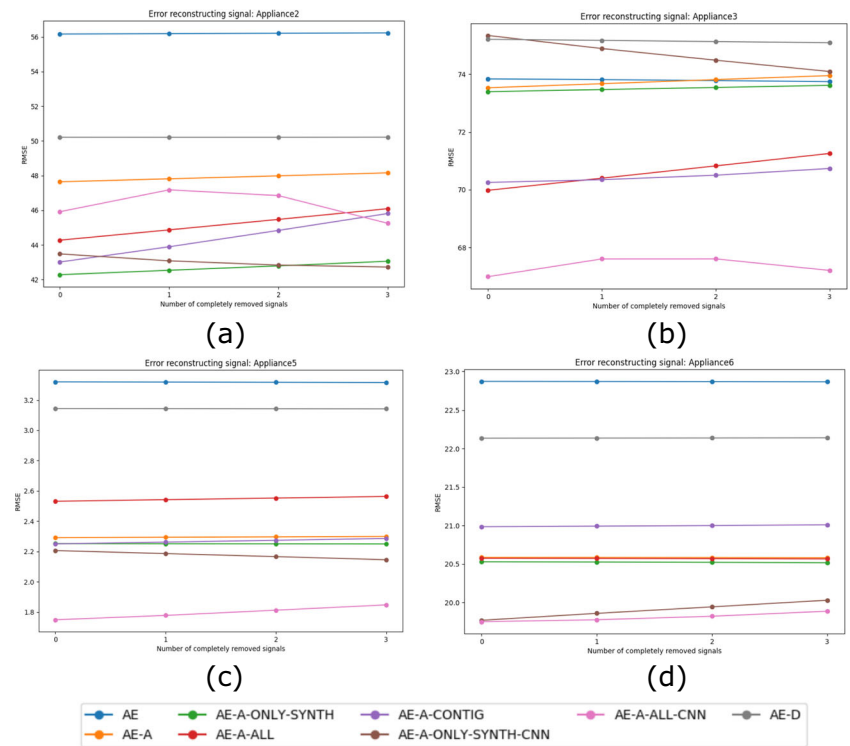


Fig. 12 Impact of other signals to the reconstruction of the signal on REFIT dataset



ing data pattern, the imputations generated by the model may also be biased or inaccurate. In the future work we will evaluate to introduce the attention mechanism in the model in order to improve the model's ability to focus on important parts of the input data that can help the imputation capability of the overall architecture.

Acknowledgements The work of Giovanni Di Gennaro has been supported by the Italian Ministry for University and Research (MUR) - PON Ricerca e Innovazione 2014-2020 (D.M. 1062/2021) The work of Francesco A.N. Palmieri and his lab has been supported by POR CAMPANIA FESR 2014/2020, ITS for Logistics, awarded to CNIT (Consorzio Nazionale Interuniversitario per le Telecomunicazioni)

Funding Open access funding provided by Ente per le Nuove Tecnologie, l'Energia e l'Ambiente within the CRUI-CARE Agreement.

Data Availability The datasets analysed during the current study are available in the related repositories as described in Section 5.

Declarations

Conflicts of interest All authors declare that they have no conflicts of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Zhao M, Kou D, Li L, Lin M (2023) An incomplete probabilistic linguistic multi-attribute group decision making method based on a three-dimensional trust network. *Appl Intell*
- Ortega J, Flórez J, Lorduy S, Jiménez G, Quintero O (2021) Improve decision-making process in air command and control systems with meteorological data fusion. In: 2021 International Conference on Decision Aid Sciences and Application (DASA), 636–642. <https://doi.org/10.1109/DASA53625.2021.9682330>
- Jiang Z, Shi D, Guo X, Xu G, Yu L, Jing C (2018) Robust smart meter data analytics using smoothed als and dynamic time warping. *Energies* 11(6). <https://doi.org/10.3390/en11061401>
- Ahn H, Kyunghye Sun KPK (2022) Comparison of missing data imputation methods in time series forecasting. *Comput Mater Continua* 70(1):767–779. <https://doi.org/10.32604/cmc.2022.019369>
- Kuppannagari SR, Fu Y, Chueng CM, Prasanna VK (2021) Spatiotemporal missing data imputation for smart power grids. In: Proceedings of the Twelfth ACM International Conference on Future Energy Systems. e-Energy '21, pp. 458–465. Association for Computing Machinery, New York. <https://doi.org/10.1145/3447555.3466586>
- Peppanen J, Zhang X, Grijalva S, Reno MJ (2016) Handling bad or missing smart meter data through advanced data imputation. In: 2016 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT), pp. 1–5. <https://doi.org/10.1109/ISGT.2016.7781213>
- Song S, Li C, Zhang X (2015) Turn waste into wealth: On simultaneous clustering and cleaning over dirty data. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '15, pp. 1115–1124. Association for Computing Machinery, New York. <https://doi.org/10.1145/2783258.2783317>
- Little R (2019) Rubin D (2019) *Statistical Analysis with Missing Data*. Wiley, Hoboken
- Fang C, Wang C (2020) *Time Series Data Imputation: A Survey on Deep Learning Approaches*. <https://doi.org/10.48550/ARXIV.2011.11347>, [arXiv:2011.11347](https://arxiv.org/abs/2011.11347)
- Che Z, Purushotham S, Cho K, Sontag D, Liu Y (2018) Recurrent neural networks for multivariate time series with missing values. *Sci Rep* 8
- Luo Y, Cai X, Zhang Y, Xu J, Xiaojie Y (2018) Multivariate time series imputation with generative adversarial networks. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds.) *Advances in Neural Information Processing Systems*, vol. 31, pp. 1603–1614. Curran Associates, Inc., Red Hook. <https://proceedings.neurips.cc/paper/2018/file/96b9bff013acedfb1d140579e2fbeb63-Paper.pdf>
- Pires IM, Garcia NM, Pombo N, Flórez-Revuelta F (2016) From data acquisition to data fusion: A comprehensive review and a roadmap for the identification of activities of daily living using mobile devices. *Sensors* 16(2). <https://doi.org/10.3390/s16020184>
- Adhikari D, Jiang W, Zhan J (2021) Imputation using information fusion technique for sensor generated incomplete data with high missing gap. *Microproc Microsys*, 103636. <https://doi.org/10.1016/j.micpro.2020.103636>
- Dasarathy BV (1997) Sensor fusion potential exploitation - innovative architectures and illustrative applications. *Proc IEEE* 85(1). <https://doi.org/10.1109/5.554206>
- Gao J, Li P, Chen Z, Zhang J (2020) A survey on deep learning for multimodal data fusion. *Neural Computation* 32(5):829–864
- Zhang P, Ma X, Zhang W, Lin S, Chen H, Yirun AL, Xiao G (2015) Multimodal fusion for sensor data using stacked autoencoders. In: 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 1–2
- Liu Z, Zhang W, Lin S, Quek TQS (2017) Heterogeneous sensor data fusion by deep multimodal encoding. *IEEE J Sel Topics Sig Proc* 11(3):479–491
- Charte D, Charte F, García S, Del Jesus MJ, Herrera F (2018) A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion* 44:78–96. <https://doi.org/10.1016/j.inffus.2017.12.007>
- Romeu P, Zamora-Martínez F, Botella-Rocamora P, Pardo J (2015) Stacked denoising auto-encoders for short-term time series forecasting. In: Koprinkova-Hristova P, Mladenov V, Kasabov NK (eds) *Artificial Neural Networks*. Springer, Cham, pp 463–486
- Liu P, Zheng P, Chen Z (2019) Deep learning with stacked denoising autoencoder for short-term electric load forecasting. *Energies* 12:2445. <https://doi.org/10.3390/en12122445>
- Wang L, Zhang Z, Chen J (2017) Short-term electricity price forecasting with stacked denoising autoencoders. *IEEE Trans Power Sys* 32(4):2673–2681
- Ryu S, Kim M (2020) Kim H (2020) Denoising autoencoder-based missing value imputation for smart meters. *IEEE Access* 8:40656–40666. <https://doi.org/10.1109/ACCESS.2020.2976500>

23. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* 11(110):3371–3408
24. Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning*. MIT Press, Cambridge
25. Ma Q, Lee W, Fu T, Gu Y, Yu G (2020) Midia: exploring denoising autoencoders for missing data imputation. *Data Mining and Knowledge Discovery*. <https://doi.org/10.1007/s10618-020-00706-8>
26. Buonanno A, Iadicicco P, Di Gennaro G, Palmieri FAN (2019) In: Esposito A. In: Faundez-Zanuy M, Morabito FC, Pasero E (eds) *Context Analysis Using a Bayesian Normal Graph*. Springer, Cham, pp 85–96
27. Di Gennaro G, Buonanno A, Palmieri FA (2021) Considerations about learning word2vec. *The Journal of Supercomputing* 77(11):1573–0484
28. Buonanno A, Nogarotto A, Cacace G, Di Gennaro G, Palmieri FAN, Valenti M, Graditi G (2021) Bayesian feature fusion using factor graph in reduced normal form. *Appl Sci* 11(4). <https://doi.org/10.3390/app11041934>
29. Buonanno A, Palmieri F (2015) Simulink implementation of belief propagation in normal factor graphs. *Smart Innovation, Systems and Technologies* 37:11–20
30. Buonanno A, Palmieri FAN (2015) Two-Dimensional Multi-layer Factor Graphs in Reduced Normal Form. In: *Proceedings of the International Joint Conference on Neural Networks, IJCNN2015, July 12-17, 2015, Killarney*
31. Di Gennaro G, Buonanno A, Palmieri FA (2021) Optimized realization of bayesian networks in reduced normal form using latent variable model. *Soft Comput* 25(10):7029–7040
32. Huang G, Liu Z, Maaten LVD, Weinberger KQ (2017) Densely connected convolutional networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. IEEE Computer Society, Los Alamitos. <https://doi.org/10.1109/CVPR.2017.243>
33. van Buuren S, Groothuis-Oudshoorn K (2011) mice: Multivariate imputation by chained equations in r. *J Sta Softw* 45(3):1–67
34. Quionero-Candela, J, Sugiyama M, Schwaighofer A, Lawrence ND (2009) *Dataset Shift in Machine Learning*. The MIT Press, Cambridge, 02142, US
35. Murray D, Stankovic L, Stankovic V (2017) An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. *Sci Data* 4:160122. <https://doi.org/10.1038/sdata.2016.122>
36. Dua D, Graff C (2017) UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Amedeo Buonanno¹  · Giovanni Di Gennaro² · Giorgio Graditi³ · Antonio Nogarotto² · Francesco A N Palmieri² · Maria Valenti¹

Giovanni Di Gennaro
giovanni.digennaro@unicampania.it

Giorgio Graditi
giorgio.graditi@enea.it

Antonio Nogarotto
antonio.nogarotto2@studenti.unicampania.it

Francesco A N Palmieri
francesco.palmieri@unicampania.it

Maria Valenti
maria.valenti@enea.it

¹ Department of Energy Technologies and Renewable Energy Sources, ENEA, P.le E. Fermi 1, 80055 Portici, NA, Italy

² Dipartimento di Ingegneria, Università degli Studi della Campania “Luigi Vanvitelli”, via Roma 29, 81031 Aversa, CE, Italy

³ Department of Energy Technologies and Renewable Energy Sources, ENEA, Via Anguillarese 301, S. Maria di Galeria, 00123 Rome, Italy