Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-10 (2025) 2230–2237

ESN-based distributed inference methods for production line energy forecasting

Andrea Bonci * **Mariorosario Prist** * **Eduard Caizer** ** **Federico Giuggioloni** ** **Lorenzo Longarini** * **Geremia Pompei** *** **Alessandro Rongoni** *

* *Department of Information Engineering Polytechnic University of Marche, Ancona, Italy, (e-mail:a.bonci@univpm.it, m.prist@staff.univpm.it, s1110740@studenti.univpm.it, s11146614@studenti.univpm.it).*

** *Department of R&D Syncode Scarl, Fermo, Italy, (e-mail: eduard.caizer@syncode.it, federico.giuggioloni@syncode.it)*

*** *Department of Computer Science University of Pisa, Pisa, Italy, (e-mail: geremia.pompei@di.unipi.it)*

Abstract: The increasing focus on environmental sustainability in industry underscores the importance of accurately predicting energy consumption to achieve efficiency goals. In this context, distributed inference proves to be a promising approach, utilizing the extensive data produced by distributed sensors in industrial environments. This study aims to create a novel methodology for precise energy consumption forecasting by integrating centralized training with distributed inference. The proposed solution has been tested on a real pilot case, a production line provided by a manufacturing company, and the results demonstrate the effectiveness of distributed inference frameworks in promoting industrial sustainability.

Keywords: Machine Learning, Energy Forecasting, Industrial Production, Distributed Learning, Echo State Network

1. INTRODUCTION

The growing focus on environmental sustainability in the industrial sector has led to an increased emphasis on optimizing energy consumption and detecting irregularities in production machinery (Reinhardt et al. (2020)). Forecasting energy consumption using Artificial Intelligence (AI) techniques is a promising field to improve the energy efficiency of industrial machinery and to reduce environmental impact. It is a critical step in transitioning to a more sustainable and environmentally conscious industry (Brito and Brito (2022)). Using AI-based approaches, the aim is to further improve energy efficiency and reduce the environmental impact of production machinery by using sensor-generated data for real-time optimization of energy consumption (Liu et al. (2024)). In fact, traditional energy forecasting methods often depend on centralized systems, which struggle with scalability and inefficiencies due to the vast and distributed nature of industrial sensor data. Centralized approaches not only introduce latency and scalability challenges, but also create bottlenecks in data transmission and raise significant privacy concerns. However, in the context of AI, state-of-the-art models, such as deep learning techniques, have shown remarkable success in forecasting tasks. However, their deployment in industrial environments faces practical challenges due to the high computational demands and resource limitations of edge devices, which are currently required to ensure the execution of controls in hard real-time (Bonci et al. (2020)). In addition, data are often collected from sensors distributed across production machines and their surrounding environments, offering a complete and real-time view of operational activities and the prevailing conditions (Rodriguez-Conde et al. (2023)). Consequently, the deployment of Distributed Learning (DL) techniques should facilitate the use of AI directly at the network edge, enabling efficient and localized data processing. In fact, Gholizadeh and Musilek (2021) reviewed the importance of applying a DL approach in the context of power systems, considering the distributed nature of their components (smart meters, generators, etc.). In this context, this study aims to develop a novel methodology for energy forecasting by combining centralized training with Distributed Inference (DI). The proposed approach applies a modified version of Echo State Networks (ESN), a lightweight recurrent neural network (Jaeger and Haas (2004)), to address the computational and scalability challenges of industrial applications and enable short-term energy forecasting to optimize energy usage with real-time operational decisions. It is valuable to consider that the proposed approach is primarily applicable to reservoir computing models that explicitly incorporate ridge regression as a readout mecha-

⋆ This research received no external funding but has been supported by Italian PON "RICERCA E INNOVAZIONE" 2014-2020 - AZIONE IV.6 "CONTRATTI DI RICERCA SU TEMATICHE GREEN"

10.1016/j.ifacol.2025.09.375

nism, such as ESNs and similar architectures, rather than to other deep neural network types (e.g., recurrent neural networks, Transformer-based models) that rely on iterative, gradient-based optimization. For these reasons, only the standard ESN and the proposed distributed ESN have been compared, avoiding comparisons with other existing approaches.

Starting from the need for scalable and efficient solutions in energy forecasting for industrial applications, previous research has already tried to explore DI methodologies. A brief overview is given below. Peccia and Bringmann (2024) have carried out a systematic review of more than one hundred papers dealing with DI on embedded and edge devices, focusing on how deep learning models can be partitioned and executed on multiple devices with limited resources. In fact, in an industrial scenario, the deep learning model training is typically performed in a centralized location, while the inference process is distributed across the network of devices (Ohlenforst et al. (2023)). Instead, Moldoveanu and Zaidi (2023) proposed a novel approach to distributed training and inference in networks with communication constraints, termed "In-Network Learning" (INL), which enables nodes in a network to collaborate in training and executing deep learning models without sharing raw data, thus preserving privacy and reducing communication overhead. The application of DL represents an effective solution to the challenges posed by resource-constrained devices (Bacciu et al. (2021) and Pompei et al. (2024)), including those used in industrial settings and the Internet of Things (IoT), as proposed by Ma et al. (2023). Recent studies have also proposed the use of paradigms such as Mobile Edge Computing (MEC) and Fog Computing (FC) to support the on-site execution of complex deep learning models, thereby promoting the adoption of lighter models that are more suitable for embedded devices (Abbas et al. (2018)). Although the paper makes reference to MEC and FC, its principal objective is to examine the potential of horizontal DI as a means to improve efficiency in IoT devices. In the aforementioned scenario, machine learning techniques, including Recurrent Neural Networks (RNN) (Schmidt (2019)) and Long Short-Term Memory (LSTM) neural networks (Hochreiter and Schmidhuber (1997)), have been extensively employed (Yadav et al. (2024)). LSTM networks have proven particularly effective even in industrial contexts for the prediction of energy consumption and anomaly management in various fields (Kermenov et al. (2023)) because of their ability to capture long-term dependencies in temporal data (Palma et al. (2024)). However, the high computational cost associated with training and inference of LSTM networks has prompted the investigation of alternative models, such as ESNs of Jaeger and Haas (2004), which offer a more resource-efficient solution while maintaining satisfactory performance in time series forecasting (Touati-Hamad and Laouar (2023)). ESNs represent an attractive alternative to energy forecasting in industry due to their simple structure and computational efficiency (Bonci et al. (2024a), and Bonci et al. (2024b)). The ESN has its deep version called Deep ESN (Gallicchio and Micheli (2020)) which is able to capture different dynamics in the various layers, allowing more complex time series to be handled. This architectural approach is particularly advantageous in scenarios where the reduction of computational load is of paramount importance without compromising accuracy of the predictions, making it an optimal choice for distributed applications on resource-constrained devices. For example, Bonci et al. (2024c), introduced an energy consumption simulation method using Deep ESN integrated with a text embedding model, which allows combining energy consumption data in production scheduling. The objective of this study is to develop a methodology for accurately forecasting energy consumption using DL techniques. This framework combines centralized training with DI, leveraging sensor networks to enable efficient, scalable, and privacy-preserving energy predictions. By addressing challenges such as computational efficiency and data transmission bottlenecks, this approach aims to enhance energy management in industrial systems. The results provide insights into the potential benefits and limitations of DL in the context of energy forecasting tasks suggesting that the proposed approach is more efficient than the classical one, without sacrificing effectiveness. Summarizing, the contributions of this paper are as follows:

- Methods for Distributed Inference Learning: two novel DL methodologies for an ESN-based model applied to the training and inference phase have been proposed. The first one proposes the Distributed Multi-Input Features (DMIF): each node processes all the data acquired in the context and exchanges information with the base station in order to set up the whole network, while the second one proposes the Distributed Single-Input Features (DSIF): each node processes only the own acquired data maintaining a more high level of privacy.
- Data Privacy: by enabling ESN-based models to be trained across multiple decentralized devices without the need to transfer raw data to a central server enhancing data privacy making it even more difficult to extract personal information from model updates. Even though the proposed approaches support data privacy, they differ from Federated Learning because the inference is computed by the central global model.
- Industrial testing setup and evaluations: a comprehensive evaluation of the proposed methods are carried out in a real industrial scenario. The dataset comprises time series data for 88 variables linked to energy parameters, collected through Industrial Smart Meters and IIoT sensors readily available on the market, facilitating replication of the experiments.

The remainder of the paper is structured as follows. Section 2 introduces the preliminaries, providing a description of the network of distributed sensors used to collect and transmit real-time environmental data for centralized processing, along with the proposed strategies. The experimental analysis, visualization of the results and discussion comparing the different performances of the three proposed strategies are highlighted in Section 3. Finally, the last Section 4 points out the conclusions and future developments.

2. MATERIALS AND METHODS

A reference scenario was considered with the aims to demonstrate the feasibility and effectiveness of DI for energy forecasting in industrial environments. It involves

a network of distributed sensor variables configured to collect environmental data in real time. The data transmitted by the sensors are then transmitted to a central base station, where they are processed to generate predictions. The processing strategies vary depending on the methods used for data collection and integration, aiming to optimize different aspects of distributed processing. A number of 88 distributed sensor variables are processed with python programming language on a Raspeberry Pi 4 with only 1 Gb RAM. The dataset, acquired from a real-world scenario consists of 88 features which are collected from 88 distributed sensor variables positioned on a production machinery. In the following sections, a detailed description of the ESN-based ML model, the proposed strategies for exploiting it, an experimental setup, and the available dataset are provided.

2.1 Echo State Network model

ESNs are a prominent method in the field of Reservoir Computing (RC) (Verstraeten et al. (2007) and Natschlger et al. (2002)), where a randomly initialized Recurrent Neural Network (RNN) acts as a dynamical system, capturing the temporal patterns of input sequences. The ESN model is extremely efficient due to the avoidance of Backpropagation Through Time (BPTT) (Werbos (1990)) to train recurrent weights. The central idea behind ESNs is that the intrinsic dynamics of the reservoir (recurrent part of ESNs) can be exploited to model complex temporal dependencies, allowing prediction of time-series data. The output of an ESN is obtained via a linear readout that is trained to map the reservoir state $\mathbf{h}(t)$ to the desired output $\mathbf{o}(t)$. The reservoir states $\mathbf{h}(t)$ are determined by applying a non-linear function f, often a hyperbolic tangent function, to the weighted sum of the input $\mathbf{x}(t)$ and previous state $\mathbf{h}(t-1)$. Mathematically, the recurrent part of the network can be described as follows:

$$\mathbf{h}(t) = f(\mathbf{W}_{in}\mathbf{x}(t) + \mathbf{W}_{rec}\mathbf{h}(t-1)) \tag{1}$$

where $\mathbf{W}_{in}$ and $\mathbf{W}_{rec}$ are the input and recurrent weights matrices respectively. The leaky integrator (Jaeger et al. (2007)) is often applied to regulate with the α hyperparameter the speed of the dynamic of the hidden states. The output can be described as follows:

$$\mathbf{o}(t) = \mathbf{W}_{out}\mathbf{h}(t) \tag{2}$$

where $\mathbf{o}(t)$ is the actual output value provided by the model, while $\mathbf{W}_{out}$ corresponds to the readout parameters, which require to be computed from a matrix $\mathbf{Y}$, the matrix of targets for each time step, and λ, i.e. the Tikhonov regularization hyperparameter (Hoerl and Kennard (1970)):

$$\mathbf{W}_{out} = \mathbf{Y}^T\mathbf{H} \cdot (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1} \tag{3}$$

In order for the Echo State Network to work properly, the reservoir should be correctly initialized. In particular, an ESN network should respect the Echo State Property analyzed in Gallicchio and Micheli (2017) and in Yildiz et al. (2012). The main method applied to respect this property is the appliance of the necessary condition:

$$\rho(\mathbf{W}_{rec}) < 1 \tag{4}$$

With this property, it is imposed that the reservoir has a matrix $\mathbf{W}_{rec}$ with spectral radius $\rho(\mathbf{W}_{rec})$ less than 1.

2.2 Strategies for distributed processing

The following section provides detailed information on the proposed strategies for distribute processing. In particular, referring to Figure 1, each strategy has been represented by an image that depicts its application pipeline, where $\mathbf{x}(t)$ indicates the input data at time t, $\mathbf{h}(t)$ denotes the internal representation generated by the model reservoir for $\mathbf{x}(t)$; $\mathbf{H}$ represents the stack of $[\mathbf{h}(1), \ldots, \mathbf{h}(T)]$ (where T is the maximum number of time-steps) provided by the each sensors; $\mathbf{o}(t)$ represents the output data for the input $\mathbf{x}(t)$. The purple cyrcles represent operators or functions: "+" is for the sum operator, "$\|$" is for concatenation operator, "Split" is for the inverse of the concatenation operator, and "RR" for the Ridge Regression learning function (Hoerl and Kennard (1970)). The concatenation operator $\|$ is able to concatenate N vectors $\mathbf{v}^n$ in another vector $\mathbf{v}$ that has the size L equal to the sum of the length l for previous N vectors:

$$\mathbf{v} = \|_{n=1}^{N} \mathbf{v}^n, \quad \text{where} \quad \mathbf{v} \in \mathbb{R}^L \wedge \mathbf{v}_n \in \mathbb{R}^l \wedge L = \sum_{n=1}^{N} l$$

In Figure 1 three main strategies can be identified, which are the following:

A) Basic centralized strategy

The Base strategy represents a traditional centralized approach to managing data generated by wireless sensors. The initial phase is the training phase, during which data is collected from various sensors and transmitted to the base station. The data is then aggregated on the base station (in Figure 1-Base, the concatenation operator is used represented by ||) to form a single input representing the global data collection. The input is then used in the generation of the internal representation $\mathbf{H}$ through the ESN reservoir. Subsequently, once the target data $\mathbf{Y}$ are available by user, ridge regression is performed to calculate $\mathbf{W}_{out}$ (3), which represents the readout of the model. At the end there is the inference phase, during which the base station receives data from the sensors in real time, concatenates the inputs, and utilizes the model to calculate the final output $\mathbf{o}(t)$ (2). This strategy centralizes both the training and the inference stages, offering a simple yet effective architectural solution.

B) Distribute Multi-Input Features strategy

The Distributed Multi-Input Features Strategy (DMIF) employs a distributed processing model. In this strategy, each sensor collects a distinct feature and transmits to all the other sensors, thereby enabling the acquisition of a multi-feature representation in each of them. Moreover, each sensor is equipped with a distinct model, which is initiated with different parameters from the others. The initial phase is the training phase, during which each sensor i, which has access to the complete set of features $\mathbf{x}(t)$, generates its own representation $\mathbf{h}^i(t)$, using the input data (1).

$$\mathbf{h}^i(t) = f(\mathbf{W}^i_{in}\mathbf{x}(t) + \mathbf{W}^i{}_{rec}\mathbf{h}^i(t-1)) \tag{5}$$

The local representations are then transmitted to the base station, which concatenates them to generate a global representation, $\mathbf{h}(t)$:

$$\mathbf{h}(t) = \|_{i=1}^{I} \mathbf{h}^i(t) \tag{6}$$

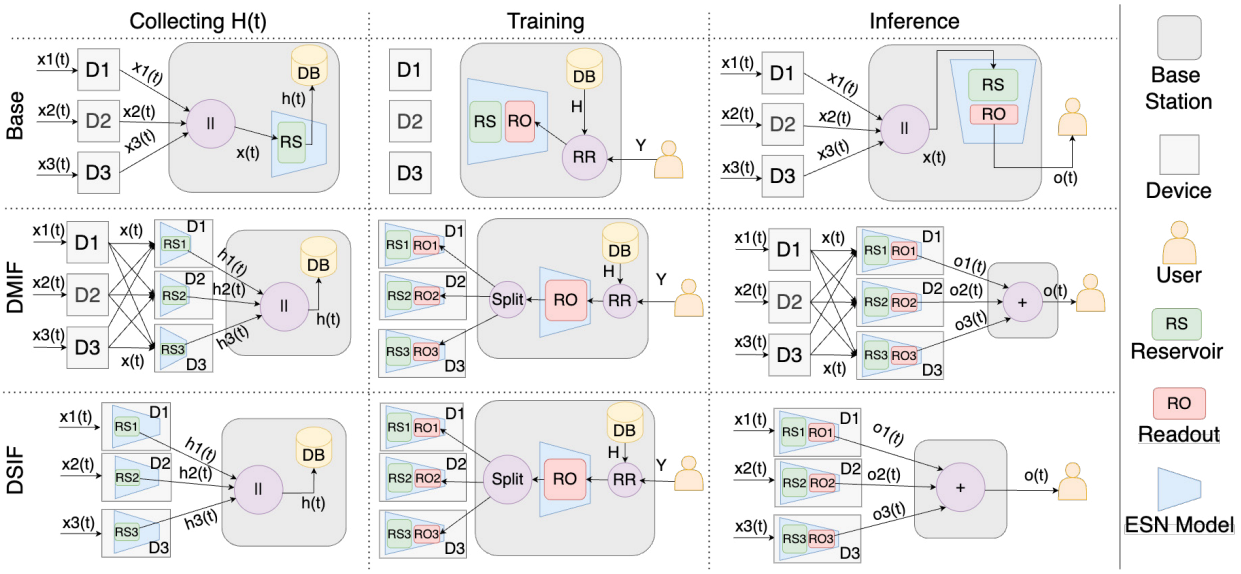

Fig. 1. Comparison of the architectures of the baseline (Base) and our proposals (DMIF and DSIF).

where I is the total number of sensors and $\|$ the concatenation operator. Subsequently, the user transmits the output targets, $\mathbf{Y}$, to the base station, which performs a global readout using ridge regression (3) with $\mathbf{H}$ that is the matrix composed by one $\mathbf{h}(t)$ for each time-step present in $\mathbf{Y}$ matrix. The resulting readout is then segmented and distributed once more to the sensors, thereby enabling each of them to possess a portion of $\mathbf{W}_{out}$:

$$\mathbf{W}_{out} = \|_{i=1}^{I} \mathbf{W}_{out}^{i} \tag{7}$$

In the inference phase, each sensor utilizes its own "mini-ESN" (comprising the preceding reservoir and the readout transmitted by the base station) to generate a partial output $\mathbf{o}^i(t)$. Subsequently, the outputs are transmitted to the base station, which aggregates them to generate the final output $\mathbf{o}(t)$. This is formalized in the following formula:

$$\begin{aligned} \mathbf{o}(t) &= \Sigma_{i=1}^{I} \mathbf{o}^i(t) = \Sigma_{i=1}^{I} \mathbf{W}_{out}^{i} \mathbf{h}^i(t) \\ &= \mathbf{W}_{out} \, \|_{i=1}^{I} \, \mathbf{h}^i(t) = \mathbf{W}_{out} \mathbf{h}(t) \end{aligned} \tag{8}$$

This approach has the effect of reducing the dependency of the base station during the processing stage, thereby optimizing scalability and network efficiency.

C) Distributed Single-Input Features strategy

The Distributed Single-Input Features strategy (DSIF) has similarities with the above introduced DMIF strategy. The principal distinction between them lies in the fact that, in this instance, a single feature is subjected to analysis, as opposed to a number of different features. Furthermore, there is no exchange of initial input data between the sensors. Similarly, each sensor is assigned different model parameters. During the training phase, each sensor i processes its data $\mathbf{x}^i(t)$ independently, maintaining own privacy:

$$\mathbf{h}^i(t) = f(\mathbf{W}_{in}^{i} \mathbf{x}^i(t) + \mathbf{W}^i{}_{rec} \mathbf{h}^i(t-1)) \tag{9}$$

Generating a representation $\mathbf{h}^i(t)$ that is sent to the base station for concatenation into a global $\mathbf{h}$ like in Formula (5) and (6). Subsequently, the ridge regression is executed on the base station and the segmented readout is distributed to the sensors, as in the previous strategy (7). During the inference phase, each sensor i employs its local ESN (which, in this case, comprises the reservoir generated during the initial phase and the readout transmitted by the base station) to generate an output, $\mathbf{o}^i(t)$, which is then transmitted to the base station. The base station then performs a total sum (8) providing $\mathbf{o}(t)$:

$$\mathbf{o}(t) = \Sigma_{i=1}^{I} \mathbf{o}^i(t) \tag{10}$$

This strategy minimizes communication between sensors, thereby simplifying the infrastructure and improving data privacy while maintaining high computational efficiency.

2.3 Experimental setup

Fig. 2. Hydraulic press machine.

The acquisition system, based on an Industrial Smart Meter, the Seneca S604 (S604 (2024)), records a multivariate vector comprising 88 different measurements, including current, voltage, power factor, displacement power factor, current harmonics, voltage harmonics, and additional variables (See Figure 3). These measurements are sampled at the frequency of 1 Hz, thereby providing a high-resolution view of the operating behavior and power consumption of machinery under a variety of conditions. Data were

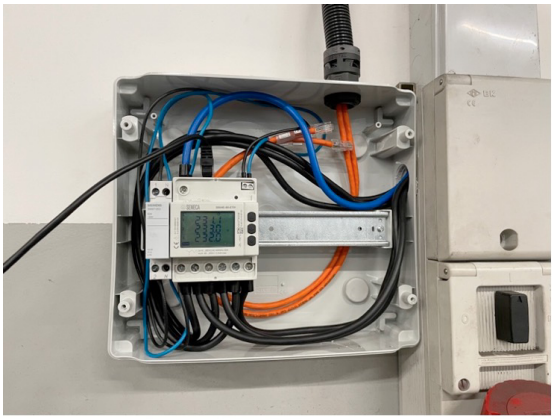

Fig. 3. Industrial Energy Consumption Smart Meter by Seneca.

collected continuously for approximately 4.5 months, resulting in a comprehensive data set that is well suited to addressing regression problems.

2.4 Dataset

The selected dataset, provides comprehensive data on the energy consumption of industrial machinery in a manufacturing environment. The main objective of this data set is to predict the energy consumption of industrial equipment based on the recorded characteristics and contextual factors. During the preparation of the data set, it has been observed that some features exhibited values close to zero when the machinery is in a state of repose. Consequently, a threshold has been applied to disregard these data. Eliminating the latter ensured that the dataset focused exclusively on active operational scenarios, thereby enhancing its relevance to the objectives of the study. To facilitate robust model evaluation, the dataset has been divided into two subsets: 80% for training and 20% for testing.

As introduced earlier, the experiments adopted a distributed approach in which each sensor has been associated with a single feature and processed independently. This configuration reduced dependency on the central station, thereby enhancing privacy and scalability. Furthermore, it exemplified a distributed processing strategy aligned with the proposed framework.

2.5 Comparison metrics

In order to properly evaluate the efficacy of the predictions of the strategies above introduced, two type of metrics have been chosen:

- Mean Absolute Error (MAE): a low MAE value is indicative of more precise predictions, as it suggests that the average discrepancy between the predicted and actual values is reduced.
- The Root Mean Square Error (RMSE): it is derived from the Mean Squared Error (MSE), a metric that weights more higher errors. A lower RMSE indicates a greater degree of efficacy in the predictions, as it implies that the discrepancies between the predicted and actual values are minimal.

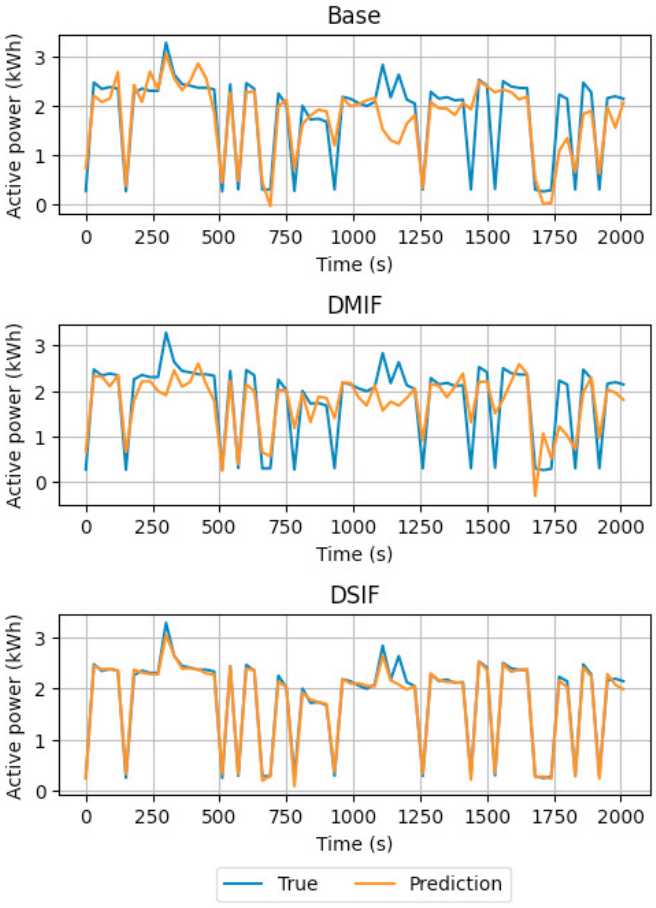

Fig. 4. Comparison between the true and predicted time series of the test set for the Base, DMIF and DSIF strategies.

3. RESULTS AND DISCUSSIONS

This section presents the analysis and commentary on the results of the experiments conducted. The analysis encompasses the identification of the optimal hyper-parameters for each strategy, which were identified through k-fold cross-validation. Once the optimal hyper-parameters were identified, they were employed to train the models of the various methodologies in a more targeted manner. Figure 4 illustrates the trend of the forecasts obtained with the three strategies, based on the energy consumption forecast dataset. The horizontal axes represent time in seconds (s), while the vertical axes depicts the active power of the system in watts (kWh). The graphs illustrate a comparison between the observed time series (blue curve) and the predicted time series (orange curve). The Base strategy demonstrates notable discrepancies from the actual curve, indicating a constrained predictive capacity. The DMIF strategy exhibits partial enhancements, yet deviations from actual values persist. Finally, the DSIF strategy displays a substantial overlap between predictions and actual data, exhibiting a discernibly more precise performance. Figure illustrates the trend of the forecasts obtained with the three strategies, based on the energy consumption forecast dataset. The horizontal axes represent the time in seconds (s), while the vertical axes depicts the active power of the system in watts (kWh). The graphs

illustrate a comparison between the observed time series (blue curve) and the predicted time series (orange curve). The DMIF strategy exhibits partial enhancements, yet deviations from actual values persist. Finally, the DSIF strategy displays a substantial overlap between predictions and actual data, exhibiting a discernibly more precise performance. The selection of the best hyperparameters for each method was performed by k-fold cross-validation with k=4, testing a range of values for each key hyperparameter (Table 1). The optimal hyperparameters include the hidden size, the regularization coefficient λ, the leakage rate, the input scaling and the spectral radius. In particular, for the hidden size parameter, values obtained by multiplying the number of features (88) by factors of 2, 5, 10 and 20 were tested. The best performance has been obtained

Table 1. Model selection best hyperparams

Hyperparam name	Base	DMIF	DSIF
hidden size	1760	20 x 88	20 x 88
λ	50	50	100
leakage rate	0.5	0.8	0.5
input scaling	0.5	0.9	0.9
spectral radius	0.99	0.99	0.99

with a hidden size of 20×88, regardless of the strategy. This choice suggests that an architecture with a higher number of hidden units allows the complex dynamics of the dataset to be captured more effectively. For the regularization coefficient λ, the DSIF strategy benefited from a higher value (100), which helped to reduce overfitting and improve model stability. The leakage rate and scaling of the inputs were also crucial in adjusting the flow of information and the sensitivity of the system to variations in the input data: the optimal leakage rate is 0.8 for DMIF and 0.5 for Base and DSIF, while the optimal scaling is 0.9 for the distributed strategies and 0.5 for Base. The spectral radius remains at 0.99 for all strategies to ensure a balance between stability and the ability to capture dynamic oscillatory behavior. The washout hyperparameter, set to 100 for all strategies, played a crucial role. This hyperparameter determines the number of initial samples that are ignored during training to reduce the influence of initial conditions. This significantly improves the efficacy and robustness of the predictions by reducing errors due to unrepresentative initial transients.

Table 2. MAE and RMSE results

Model	MAE	RMSE
Base	0.36 ±0.05	0.479 ±0.05
DMIF	0.37 ±0.06	0.48 ±0.066
DSIF	**0.059** $\pm 9 \times 10^{-4}$	**0.136** $\pm$ 1.6×10^{-3}

After identifying the best hyperparameters, the models were retrained to maximise performance. The results for MAE and RMSE, presented in Table 3, were calculated as the mean over 20 iterations, with their respective standard deviations (indicated by ± next to the result). The DSIF strategy performs significantly better with an average MAE of 0.059±0.0009 and an average RMSE of 0.136±0.0016. These values are significantly better than DMIF (MAE=0.37 and RMSE=0.48) and the baseline strategy (MAE=0.36 and RMSE=0.479). This indicates that the DSIF approach is able to capture the dynamics of the time series more accurately, thus providing more accurate and robust predictions than the other strategies tested. In addition to the evaluation of forecast accuracy, the execution times for data collection, training, and inference have been analyzed in Table 3. These results demonstrate that distributed approaches achieve significant reductions in execution time compared to the centralized Base strategy.

Table 3. Execution times.

Model	collection (s)	training (s)	inference (s)
Base	3.49 ±0.27	2.15 ±0.10	0.69 ±0.08
DMIF	2.4 ±0.33	2.14 ±0.17	0.49 ±0.04
DSIF	**2.33** ±0.22	**2.11** ±0.07	**0.40** ±0.02

4. CONCLUSION

This study illustrates the considerable potential of distributed deep learning methodologies to enhance the prediction of energy consumption in industrial machinery. The architectural approach investigated in this study is based on the ESN model, which has been developed through three distinct strategies: the Basic centralized, the Distributed Multi-Input Features (DMIF), and the Distributed Single-Input Features (DSIF). The DSIF strategy demonstrated superior performance compared to both the Base and DMIF strategies, with significantly lower error metrics. The capacity to process single-input features in a distributed context enabled the model to achieve accurate predictions. The Base centralized strategy, despite its simplicity and centralized architecture, demonstrated greater discrepancies, indicating its limited scalability. In contrast, the DMIF strategy exhibited improvements over the latter, but less effective than DSIF due to the increased complexity and interdependencies introduced by the exchange of multi-input features. The application of distributed ESN is in line with industrial sustainability goals, as it enables accurate energy forecasting, facilitating optimized energy consumption, reduced waste, and the minimization of the environmental impact of manufacturing operations. These results demonstrate the value of DI frameworks in promoting industrial sustainability. While the preliminary findings are promising, further exploration is required, particularly in areas such:

- Combine both DMIF and DSIF methodologies to further optimize predictive accuracy and scalability in complex industrial environments.
- Extend the analysis from a single production line to an entire production area, considering each production line as a sensor.

ACKNOWLEDGEMENTS

Special acknowledgment is given to Sifim Srl for their cooperation in installing a smart meter sensor on their production machine, and heartfelt thanks are offered to Syncode Scarl for their invaluable support in a major research project, including the provision of essential hardware devices and software infrastructure.

REFERENCES

Abbas, N., Zhang, Y., Taherkordi, A., and Skeie, T. (2018). Mobile edge computing: A survey. *IEEE*

Internet of Things Journal, 5(1), 450–465. doi: 10.1109/JIOT.2017.2750180.

Bacciu, D., Di Sarli, D., Faraji, P., Gallicchio, C., and Micheli, A. (2021). Federated reservoir computing neural networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–7. doi: 10.1109/IJCNN52387.2021.9534035.

Bonci, A., Fredianelli, L., Kermenov, R., Longarini, L., Longhi, S., Pompei, G., Prist, M., and Verdini, C. (2024a). Deepesn neural networks for industrial predictive maintenance through anomaly detection from production energy data. *Applied Sciences*, 14(19). doi: 10.3390/app14198686.

Bonci, A., Kermenov, R., Longarini, L., Longhi, S., Pompei, G., Prist, M., and Verdini, C. (2024b). An echo state network-based light framework for online anomaly detection: An approach to using ai at the edge. *Machines*, 12(10). doi:10.3390/machines12100743.

Bonci, A., Longhi, S., Nabissi, G., and Scala, G.A. (2020). Execution time of optimal controls in hard real time, a minimal execution time solution for nonlinear sdre. *IEEE Access*, 8, 158008–158025. doi: 10.1109/ACCESS.2020.3019776.

Bonci, A., Prist, M., Pompei, G., Longarini, L., Biase, A.D., and Verdini, C. (2024c). Deep learning and text-embedding to integrate energy consumption into industrial machine production planning. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–4. doi: 10.1109/ETFA61755.2024.10710763.

Brito, T.C. and Brito, M.A. (2022). Forecasting of energy consumption : Artificial intelligence methods. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–4. doi: 10.23919/CISTI54924.2022.9820078.

Gallicchio, C. and Micheli, A. (2017). Echo state property of deep reservoir computing networks. *Cognitive Computation*, 9. doi:10.1007/s12559-017-9461-9.

Gallicchio, C. and Micheli, A. (2020). Deep echo state network (deepesn): A brief survey. *CoRR*. doi: 0.48550/arXiv.1712.04323.

Gholizadeh, N. and Musilek, P. (2021). Distributed learning applications in power systems: A review of methods, gaps, and challenges. *Energies*, 14(12).

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9, 1735–80. doi: 10.1162/neco.1997.9.8.1735.

Hoerl, A.E. and Kennard, R.W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55–67. doi: 10.1080/00401706.1970.10488634.

Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304, 78 – 80.

Jaeger, H., Lukoševičius, M., Popovici, D., and Siewert, U. (2007). Optimization and applications of echo state networks with leaky- integrator neurons. *Neural Networks*, 20(3), 335–352. doi: https://doi.org/10.1016/j.neunet.2007.04.016.

Kermenov, R., Nabissi, G., Longhi, S., and Bonci, A. (2023). Anomaly detection and concept drift adaptation for dynamic systems: A general method with practical implementation using an industrial collaborative robot. *Sensors*, 23(6). doi:10.3390/s23063260.

Liu, S., Xiang, Y., and Zhou, H. (2024). A deep learning-based approach for high-dimensional industrial steam consumption prediction to enhance sustainability management. *Sustainability*, 16(22).

Ma, T., Wang, H., and Li, C. (2023). Quantized distributed federated learning for industrial internet of things. *IEEE Internet of Things Journal*, 10(4), 3027–3036. doi:10.1109/JIOT.2021.3139772.

Moldoveanu, M. and Zaidi, A. (2023). In-network learning: Distributed training and inference in networks. *Entropy*, 25(6). doi:10.3390/e25060920.

Natschlger, T., Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states:.

Ohlenforst, T., Schreiber, M., Kreyß, F., and Schrauth, M. (2023). Enabling distributed inference of large neural networks on resource constrained edge devices using ad hoc networks. In S. Ossowski, P. Sitek, C. Analide, G. Marreiros, P. Chamoso, and S. Rodríguez (eds.), *Distributed Computing and Artificial Intelligence, 20th International Conference*, 145–154. Springer Nature Switzerland, Cham.

Palma, G., Chengalipunath, E.S.J., and Rizzo, A. (2024). Time series forecasting for energy management: Neural circuit policies (ncps) vs. long short-term memory (lstm) networks. *Electronics*, 13(18). doi: 10.3390/electronics13183641.

Peccia, F.N. and Bringmann, O. (2024). Embedded distributed inference of deep neural networks: A systematic review. *CoRR*. doi:10.48550/arXiv.2405.03360.

Pompei, G., Dazzi, P., De Caro, V., and Gallicchio, C. (2024). Decentralized incremental federated learning with echo state networks. In *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. doi: 10.1109/IJCNN60899.2024.10650756.

Reinhardt, H., Bergmann, J.P., Münnich, M., Rein, D., and Putz, M. (2020). A survey on modeling and forecasting the energy consumption in discrete manufacturing. *Procedia CIRP*, 90, 443–448. doi: https://doi.org/10.1016/j.procir.2020.01.078.

Rodriguez-Conde, I., Campos, C., and Fdez-Riverola, F. (2023). Horizontally distributed inference of deep neural networks for ai-enabled iot. *Sensors*, 23(4). doi: 10.3390/s23041911.

S604 (2024). Seneca smart meter. URL `www.seneca.it/media/4166/power_1912eng_r1.pdf`. Accessed on December 12th, 2024.

Schmidt, R.M. (2019). Recurrent neural networks (rnns): A gentle introduction and overview. *CoRR*. doi: 10.48550/arXiv.1912.05911.

Touati-Hamad, Z. and Laouar, M.R. (2023). Predicting energy consumption through deep learning-powered time series analysis. In *4th International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2023)*, volume 2023, 424–434. doi:10.1049/icp.2024.0522.

Verstraeten, D., Schrauwen, B.U., D'Haene, M., and Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, 20(3), 391–403. doi: https://doi.org/10.1016/j.neunet.2007.04.003.

Werbos, P. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10), 1550–1560. doi:10.1109/5.58337.

Yadav, S., Bailek, N., Kumari, P., Nuţă, A.C., Yonar, A., Plocoste, T., Ray, S., Kumari, B., Abotaleb, M., Alharbi, A.H., Khafaga, D.S., and El-Kenawy, E.S.M. (2024). State of the art in energy consumption using deep learning models. *AIP Advances*, 14(6), 065306.

Yildiz, I.B., Jaeger, H., and Kiebel, S. (2012). Re-visiting the echo state property. *Neural Networks*, 35, 1–9. doi: https://doi.org/10.1016/j.neunet.2012.07.005.