Quantum and Quantum-Inspired architectures for Efficient Machine Learning

Guillermo Rubiños Rodríguez[1,4][0000−0002−7707−7773], Martín Ottavianelli[1][0009−0002−9299−7970], Pablo Díez-Valle[1][0000−0001−8338−7973], Daniel Manzano[2,3][0000−0001−9241−5884], and Tomás F. Pena[4,5][0000−0002−7622−4698]

[1] Instituto Tecnológico de Galicia, Cantón Grande 9, Planta 3, 15003 A Coruña, Spain.
grodriguez@itg.es
[2] Quantum Thermodynamics and Computation Group. Departamento de Electromagnetismo y Física de la Materia, Universidad de Granada, 18071 Granada, Spain.
[3] Instituto Carlos I de Física Teórica y Computacional, Universidad de Granada, 18071 Granada, Spain.
[4] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, 15782, Spain
[5] Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Santiago de Compostela, 15782, Spain

Abstract. The search for new energy-efficient algorithms has recently gained attention as the energy consumption of machine learning (ML) workloads continues to increase. In this article, we provide an initial comparison between two efficient ML models based on quantum systems. The first approach is a quantum reservoir computing (QRC) model based on driven spin dynamics, while the second approach is a parameterized tensor network (TN) model, specifically a Matrix Product State (MPS) architecture. These models are evaluated on two forecasting tasks of a different nature: autonomous forecasting of the chaotic Mackey–Glass time series, and direct hourly electrical-load forecasting. The QRC model exhibits lower execution times while preserving competitive predictive performance in both tasks, highlighting its energy efficiency; whereas the MPS model provides greater stability across random seed initializations while also achieving accurate forecasts. The results show that the proposed methods are efficient quantum and quantum-inspired machine learning alternatives for time series forecasting. We expect this comparative study to stimulate further development, scaling and benchmarking against a broader range of more complex ML tasks.

Keywords: Quantum Reservoir Computing · Tensor Networks · Time series forecasting · Quantum Machine Learning.

1 Introduction

The amount of data generated and processed using machine learning (ML) models is increasing rapidly. Recent studies suggest that ML-related electricity demand could double by 2030, reaching approximately 945 TWh (slightly exceeding Japan's energy consumption today) [6]. This growth is driven not only by the increasing volume of data being stored, but also by the large-scale ML models currently in use, which require substantial computational resources, energy consumption, and training time. In this context, the search for efficient quantum machine learning (QML) algorithms that consume less energy, memory and time has attracted increasing attention.

In this article, our main contribution is a comparison of two methods based on quantum systems and their intrinsic properties applied to time series forecasting. We evaluate the proposed methods on two tasks of a fundamentally different nature: one derived from a chaotic mathematical system, and the other based on a real-world energy consumption dataset. We do not aim to compare them to classical state-of-the-art baselines. The proposed methods were designed to offer a tradeoff between performance, parameter count and training time, rather than to be optimal on any single one of these axes, and an unbiased comparison is beyond the scope of this article.

The first model is a spin-based Quantum Reservoir Computing (QRC) algorithm. This paradigm is a supervised neuromorphic algorithm that uses a quantum system complex evolution to produce a non-linear high-dimensional representation of the data whose measurements are later processed using an efficient classical linear regression [16].

The second method is a Matrix Product State (MPS) model, a widely used Tensor Network (TN) architecture [4]. TN methods comprise a collection of architectures and techniques that enable the decomposition of high-order tensors into networks of lower-rank tensors, and have recently found applications in ML, giving rise to the field of Tensor Network Machine Learning (TNML).

This article is organized as follows. In Sec. 2 we describe the selected QRC and TN models. In Sec. 3 we present and compare the results obtained for the two ML tasks considered. Finally, conclusions and future research directions are discussed in Sec. 4.

2 Models

2.1 Quantum Reservoir Computing

QRC extends the classical reservoir computing framework to quantum systems. This paradigm uses a complex dynamical evolution that maps inputs into a high dimensional feature space. The exponential Hilbert space and the highly complex non-linear dynamics characteristic of many quantum systems present them as an interesting platform for this framework. In recent years, these algorithms

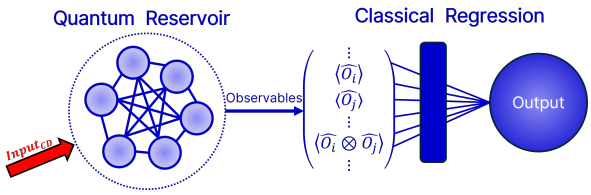


Fig. 1: **Quantum Reservoir Computing illustration for an all connected Ising spin system with continuous driving as input method.** The input is encoded through a local coherent driving on all qubits, then the system is evolved for a time dt and afterwards a set of observables is measured. The measured outcomes are used to predict the output using an efficient classical regression.

have attracted increasing attention as an alternative to variational quantum algorithms for QML applications because they do not suffer from vanishing gradients (barren plateaus).

In this article, we implement the continuously driven fully connected spin Ising model of $N = 5$ qubits proposed in ref. [21], whose Hamiltonian is

$$H(s_k) = \sum_{i>j} J_{i,j}\, \sigma_i^x \sigma_j^x + \sum_i h_i \sigma_i^z + s_k \sum \sigma_i^x \,, \tag{1}$$

where $s_k \in [0, 1]$ is the k-th value of the normalized input series $\boldsymbol{s} = (s_0, s_1, ..., s_l)$, σ_i^a $(a = x, y, z)$ are the Pauli matrices on the i-th qubit, $J_{i,j}$ are the coupling weights between atoms i, j and h_i is the external magnetic field related to site i . Local dissipation is also considered for this system, resulting in a dynamical map defined by the following Gorini-Kossakowski-Lindblad-Sudarshan (GKLS) Master equation [12]

$$\frac{d\rho}{dt} = -i\,[H, \rho] + \gamma \sum_{i=1}^{N} \left(\sigma_i^- \rho \sigma_i^+ - \frac{1}{2}\{\sigma_i^+ \sigma_i^-, \rho\} \right), \tag{2}$$

where γ is the local decay rate of the spins. The optimal hyperparameters found in [21] are: $dt = 0.1$, $\gamma = 10$, $h = 0.1$ and $J_{i,j}$ are uniformly sampled from $\left[-\frac{1}{2}, \frac{1}{2}\right]$.

As depicted in Fig. 1, each input of data is evolved following equation 2 and at the end of this evolution a selected set of observables is measured. These measured observables are appended to a matrix $\boldsymbol{x}$ that contains the measured results for inputs of the training set of the dataset. This vector is then used to train a linear regression that performs the desired task.

This proposed model has the *universal approximation property* if the following conditions are satisfied [21]: echo state property (ESP) [27], fading memory property (FMP) [5], and input separability.

Why should we use QRC for ML applications? On the one hand, QRC has very few trainable parameters and a fast inference and training process performed by an **efficient** classical regression. Moreover, it requires very

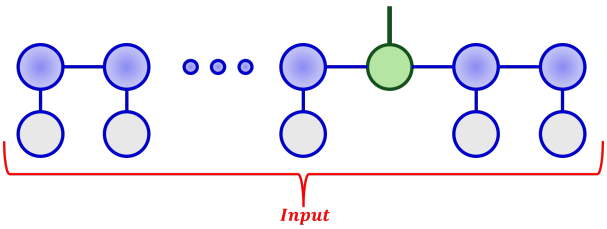


Fig. 2: **Matrix Product State (MPS) representation with open boundary conditions.** Each site is associated with a tensor containing one physical index of dimension d and up to two virtual (bond) indices of dimension D. The bond indices connect neighboring tensors and encode the correlations captured by the network, whereas the physical indices represent the local feature spaces associated with the input data. In addition to the input tensors, the network includes an output tensor (in green) that maps the encoded representation to the target prediction.

low **memory** to just store the trained parameters of the regression. On the other hand, the dynamical and quantum properties of the selected reservoir let us **interpret** how the information is being processed. As an example, the ergodic chaotic dynamical regime has been found to be better than the many-body-localized phase for QRC [13]. Apart from classical state-of-the-art tools to **explain** the predictions of the model, the use of quantum information tools is an interesting alternative that may provide some insight into information or correlations that classical tools are not able to obtain.

2.2 Tensor Networks

Tensor Networks have recently emerged as a promising alternative to state-of-the-art neural networks for developing computationally efficient and interpretable ML models [25]. TNs provide compact representations that reduce the number of trainable parameters while preserving expressive power, offering a scalable and computationally efficient framework for ML, with quantum information theory providing valuable tools to analyze and interpret learned representations [20].

TNs comprise several architectures distinguished by their connectivity patterns and correlation structures. Among the most widely used in ML are MPS, a one-dimensional tensor network (see Figure 2) whose expressive power is controlled by the bond dimension D, which determines the amount of correlations that can be represented (analogous to the role of entanglement in quantum many-body systems)[19]. Although exact representations may require exponentially large bond dimensions, many physical systems and ML datasets exhibit structured correlations that can be captured accurately with relatively small values of D. Recent work has demonstrated the applicability of TNs to supervised [24], unsupervised [23], and reinforcement learning [14], as well as neural-network

compression [17]. In tasks such as anomaly detection [26], sequence modeling [15], and constrained generative modeling [10], MPS-based models have achieved competitive, and in some cases superior, performance compared with conventional approaches.

Why should we use TN for ML applications? The analogous growth in the dimensionality of vector spaces encountered in both quantum many-body physics and ML has motivated the use of tensor networks as compact representations of high-dimensional data, improving the trade-off between model expressiveness and computational cost. By mitigating the "curse of dimensionality", TNs can substantially reduce the number of trainable parameters, leading to lower computational and energy requirements during both training and inference. In the case of MPS, a key ingredient underlying this **efficiency** is the existence of canonical forms and the use of singular value decompositions (SVDs). By truncating the smallest singular values, MPS can be compressed through optimal low-rank approximations, reducing the bond dimension and the number of model parameters while preserving the most relevant information encoded in the data. Although some information may be discarded during this process, SVD-based truncation has been shown to provide the optimal compression strategy [3]. Beyond efficiency, TNML also offers a promising route toward more **interpretable** ML models. While a significant performance gap often exists between interpretable probabilistic ("white-box") approaches and state-of-the-art deep neural networks, tensor networks provide access to a rich mathematical framework inherited from quantum information theory. Quantities such as reduced density matrices, von Neumann entropy, and mutual information can be employed to characterize correlations, quantify information content, and analyze the internal representations learned by the model [20,1]. As a result, TNs not only enable efficient representations of complex datasets, but also provide principled tools for understanding how information is encoded and processed within the network.

3 Results and Discussion

We begin by introducing the benchmark tasks. Next, we describe the model configurations adopted for each task. Finally, we present and discuss the results obtained by the different approaches.

3.1 Tasks

We test both models on predicting the Mackey-Glass (MG) [11] chaotic time series and the PJM energy consumption dataset [9].

Mackey-Glass. The MG chaotic time series is a widely used benchmark for time series predictions in both ML and QML [18]. The MG function $u(t)$ is obtained by numerically solving the following delay-differential equation

$$\frac{du(t)}{dt} = -0.1u(t) + \frac{0.2u(t-\tau)}{1+u^{10}(t-\tau)}. \tag{3}$$

This delay-differential equation generates chaotic dynamics when $\tau > 16.8$ [11], therefore we use $\tau = 17$ as most of the studies regarding QRC. We remark that t is dimensionless. The values are obtained by numerically solving equation 3 and sampling the values using a time resolution t_r of 3. We discard the first 1000 values, use the next 1000 as training and the following 150 as test. The aim of this task is to perform **autonomous recursive prediction**, where the predicted value is used as input for the next step in the test phase.

PJM Energy Consumption database. This database contains time-series describing the hourly electricity consumption of the Eastern Interconnection grid, as reported by PJM Interconnection LLC [9]. Unlike the Mackey–Glass benchmark, these series exhibit less chaotic dynamics. For simplicity, we focus exclusively on the PJME time series, one of the many datasets within this database. The objective of this task is to evaluate the ability of the models to predict electricity demand 24 hours in advance.

The selected metrics to compare the results are the normalized mean absolute error (**NMAE**), the normalized mean squared error (**NMSE**), the normalized root mean squared error (**NRMSE**), the trainable **parameter count** and the training time.

3.2 Ising Quantum Reservoir

We expand the results of the reference article [21] by considering the following observable sets:

- $\mathcal{O}_1 = \langle \sigma_i^a \rangle \bigcup \langle \sigma_i^a \sigma_j^a \rangle$, with $a = x, y, z$ and $1 \leq i, j \leq N$, $i \neq j$. Used in the reference article.
- $\mathcal{O}_2 = \langle \sigma_i^a \rangle \bigcup \langle \sigma_i^a \sigma_j^b \rangle$, with $a, b = x, y, z$ and $1 \leq i, j \leq N$.

The presented results are obtained by integrating the GKLS equation 2 using the *mesolve* function from the QuTip package in Python [8].

The autonomous prediction method for **Mackey-Glass** enables us to study not only how the reservoir learns the internal dynamics of the time series, but also how it behaves against the accumulated error that expands through the prediction. We depict the results in Figure 3 and in Table 1. Since the defined reservoir depends on random sampling from a uniform distribution, we averaged over 100 random initializations of the reservoir.

The larger observable set ($\mathcal{O}_2$) returns the best (lowest) mean result in all prediction metrics at the cost of more training and execution time compared to the $\mathcal{O}_1$ set.

The direct prediction method implemented for the **PJME** dataset uses one input to predict the energy consumption 24 hours later. The large size of the dataset and the stochastic nature of the reservoir make it necessary to average over seeds to extrapolate reliable results. Consequently, training the QRC model

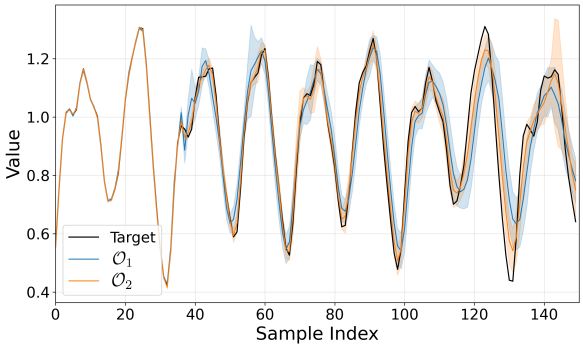


Fig. 3: **Autonomous prediction of the MG timeseries for the QRC model.** The predictions for both observable sets are averaged over 100 random initialization of the reservoir. The shaded regions cover the one standard deviation taken as statistical measure.

using large percentages of the dataset is not viable within a reasonable execution time. Alternatively, we randomly select a starting month from the dataset and use it as washout period, allowing it to satisfy the necessary conditions for universality [22]. The next 10 or 15 months are used for training, after which the model directly predicts each of the 24 hourly load values for the following day, one horizon at a time, rather than recursively.

In this way, we compare two observable sets and two different training sizes. The results are presented in table 1 and a visualization of the $\mathcal{O}_1$ prediction for both training sizes is displayed along the MPS predictions in figure 5. As expected, the biggest training size yields better results for both observable sets. In contrast with the results obtained for the MG task, we find that the $\mathcal{O}_1$ observable set performs better for the direct prediction setting.

3.3 Tensor Networks

The proposed model for both MG and PJME consists of a sliding-window MPS architecture. The window size and bond dimension were determined through a hyperparameter search aimed at maximizing predictive accuracy while maintaining feasible training and inference costs for each task.

The specific model implemented for **Mackey-Glass** prediction employs a window size of 227, corresponding to an MPS with 227 input tensors, 1 output tensor, and a bond dimension of 100. A linear local feature map is used to encode the inputs, mapping each time-series value x onto the three-dimensional vector $(1, x, 1-x)$. The physical dimension of the feature map, together with the selected bond dimension and window size, leads to a comparatively large number of trainable parameters. Nevertheless, the relatively small size of the

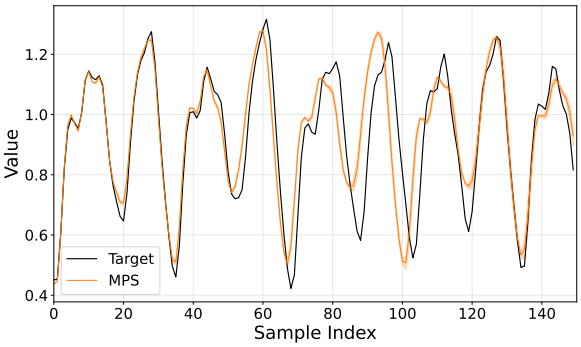


Fig. 4: **Autonomous prediction of the MG time series using the MPS model.** Predictions are averaged over 10 random initializations of the MPS. The barely visible shaded regions represent one standard deviation.

dataset (1000 training samples and 300 test samples) keeps both training and inference times within reasonable limits.

We observed little variation across random seeds in the recursive forecasting task; therefore, all reported results were averaged over only 10 independent random initializations. As shown in Table 1, the MPS model achieves predictive performance comparable to that of the QRC approach, with an NRMSE of approximately 0.1. Nevertheless, this performance comes at the cost of a considerably larger model size and parameter count. This reflects the complexity of the Mackey–Glass forecasting problem, which requires the model to capture long-range temporal correlations and highly nonlinear dynamics in order to achieve accurate predictions from a relatively small dataset. The autonomous prediction produced by the MPS model is illustrated in Figure 4.

For the **PJME** forecasting task, there are two key differences with respect to the QRC-based approach. First, the full dataset is utilized in this setting, with the data split into training, validation, and test sets comprising 75%, 12.5%, and 12.5% of the samples, respectively. Second, and perhaps most importantly when interpreting the reported performance metrics, the MPS model is trained to predict the entire 24-hour load profile, whereas the QRC-based approach predicts only a single value corresponding to the load 24 hours ahead. This distinction substantially changes the forecasting task and should be taken into account when comparing the results obtained with both approaches.

Here, we employ a sliding-window MPS architecture with a window size of 168 sites, corresponding to one week of hourly electrical load measurements. The bond dimension was also selected through hyperparameter optimization and set to 16. Although larger bond dimensions could be considered, the size of the dataset makes them significantly more demanding in terms of both training and inference costs, while providing only limited performance improvements. In addition, a quadratic local feature map is used to encode the inputs, mapping each value x onto the three-dimensional vector $(1, x, x^2)$.

Table 1: **Results for both considered tasks and ML models.** We remark that the PJME MPS results are not averaged over seeds because of its stability.

			N-RMSE	N-MSE	N-MAE	#PARAMS
MG	**MPS**		0.1001 ± 0.0083	0.0101 ± 0.0017	0.0635 ± 0.0082	6820200
	QRC $\mathcal{O}_1$		0.145 ± 0.065	0.025 ± 0.023	0.096 ± 0.044	45
	QRC $\mathcal{O}_2$		0.095 ± 0.084	0.016 ± 0.039	0.056 ± 0.044	105
PJME	**MPS Baseline**		0.040	0.0017	0.029	135200
	MPS Compressed		0.041	0.0018	0.030	116318
	QRC 10 month	$\mathcal{O}_1$	0.036 ± 0.026	0.0020 ± 0.0033	0.031 ± 0.024	45
		$\mathcal{O}_2$	0.050 ± 0.032	0.0035 ± 0.0046	0.041 ± 0.028	105
	QRC 15 month	$\mathcal{O}_1$	0.031 ± 0.019	0.0013 ± 0.0017	0.027 ± 0.018	45
		$\mathcal{O}_2$	0.041 ± 0.022	0.0023 ± 0.0024	0.036 ± 0.020	105

The results obtained with the MPS model are presented in Table 1. Although these results are not competitive with the current state-of-the-art in short-term electrical load forecasting (see [2] for instance), they remain competitive with a range of classical ML approaches (see [7]). Moreover, this performance is achieved using a relatively compact model, highlighting the ability of TNs to provide efficient representations with reduced memory requirements and parameter count.

We further compress the model by applying an SVD-based truncation scheme to reduce the bond dimension and, consequently, the number of network parameters. Although this compression leads to some degradation in predictive performance, it provides an interesting opportunity to explore the trade-off between model accuracy, memory requirements, and computational cost. The results obtained with the SVD-compressed model are also reported in Table 1. This compression was performed by retaining the singular values accounting for 99% of the total singular-value weight, after which the model was fine-tuned for an additional five epochs. Figure 5 presents the predictions of all considered models. The predictions labeled "MPS retrain" are the compressed and fine-tuned MPS.

3.4 Discussion

The models under comparison are fundamentally different: QRC is a neuromorphic computing method while the MPS approach is closer to conventional neural-network models, both in terms of its architecture and its training procedure. In both tasks the MPS model contains substantially more parameters than the QRC model. In the following discussion we take this into account as well as the previously defined metrics in order to draw conclusions.

We compare the overall results for both tasks in Table 1. For MG, both models achieve similar performance metrics, so it could be said that they appear to capture the underlying dynamics of this chaotic time series. The MPS model shows stable behavior, with low variability across random initializations. As expected in the QRC method, the larger observable set yields the best performance, indicating that a richer readout from the system improves autonomous

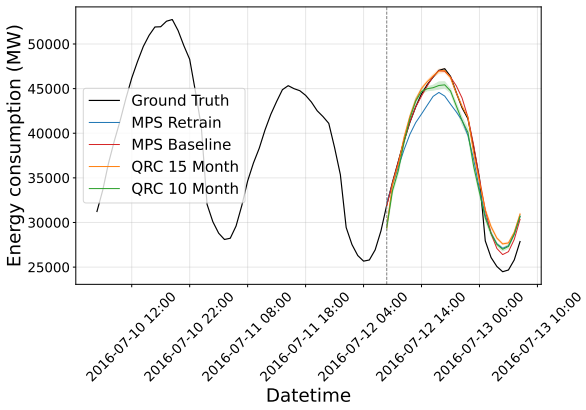


Fig. 5: **24 hour predictions for the PJME task for the MPS and QRC model.** We include the two MPS models and the two considered training sizes for the $\mathcal{O}_1$ observable set.

prediction. Another observed result is that the QRC execution time (simulated training and inference) is shorter than the MPS execution time. This increased computational cost is consistent with the considerably larger number of parameters employed by the MPS architecture for this task. The mean execution time for simulated QRC is around 45 seconds, while the mean execution time for the MPS model exceeds 110 seconds.

Table 1 also displays the best results for the direct prediction of the PJME dataset. A direct comparison between both architectures should be avoided, since the training and inference protocols and the evaluation methodology are not equivalent: QRC results are averaged over 50 randomly selected time periods and the model predicts only a single value corresponding to the load 24 hours ahead (24-hour-ahead-forecasting), whereas the MPS uses 75% of the dataset for training and aims to predict the entire 24-hour load profile (24-step forecasting). Both models perform well within their respective forecasting settings. The QRC model's high standard deviation suggests that the prediction difficulty depends on the selected time window, possibly due to seasonal effects or abrupt changes in demand. The TN addresses a more demanding forecasting task while its structure enables systematic compression through SVD-based truncation, allowing the bond dimension and parameter count to be reduced with only a limited impact on predictive accuracy.

4 Conclusion

This work explores two distinct ML approaches inspired by quantum systems, briefly reviewing the state-of-the-art and examining their predictive capabilities, computational efficiency, and interpretability and explainability properties. We evaluate these models on two forecasting tasks of interest: MG chaotic time-series

prediction and hourly electrical load forecasting (PJME), obtaining comparable predictive performance with both approaches. Specifically, we found the QRC paradigm to be more efficient, both in terms of parameter count and training time, than the MPS model, while the MPS model exhibits greater stability across random initializations and appears better suited to learning from larger datasets.

While this modest reservoir worked quite well on these tasks, the implementation of larger reservoirs and their benchmark against other datasets is an interesting study to conduct in the future. Additionally, the study of quantum properties as explainability tools for time series prediction is an intriguing question.

For TNML, future work could systematically analyze the trade-off between bond dimension, model size, and predictive performance for forecasting tasks; explore more expressive architectures (such as Tree Tensor Networks); and leverage quantum-information-theoretic tools to better understand learned representations and use these insights to design better architectures.

Acknowledgments. We acknowledge project PID2024-162155OB-I00, funded by MICIU/AEI/10.13039/501100011033/ FEDER, UE, as well as the QUANTUM ENIA project call (Quantum Spain project) funded by the Ministry of Economic Affairs and Digital Transformation of the Spanish Government. This research has been supported by ARQADE (CER-20251019), funded by the CERVERA Research Programme of CDTI (Centre for Technological Development and Innovation).

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Aizpurua, B., Palmer, S., Orús, R.: Tensor networks for explainable machine learning in cybersecurity. Neurocomputing **639**, 130211 (2025)
2. Alanazi, M.D., Saeed, A., Islam, M., Habib, S., Sherazi, H.I., Khan, S., Shees, M.M.: Enhancing short-term electrical load forecasting for sustainable energy management in low-carbon buildings. Sustainability **15**(24), 16885 (2023)
3. Catarina, G., Murta, B.: Density-matrix renormalization group: a pedagogical introduction. European Physical Journal B **96**, 111 (2023)
4. Cirac, J.I., Pérez-García, D., Schuch, N., Verstraete, F.: Matrix product states and projected entangled pair states: Concepts, symmetries, theorems and applications. Reviews of Modern Physics **93**, 045003 (2021)
5. Del Papa, B., Priesemann, V., Triesch, J.: Fading Memory, Plasticity, and Criticality in Recurrent Networks. Springer International Publishing (2019)
6. International Energy Agency: Artificial intelligence, `https://www.iea.org/topics/artificial-intelligence`
7. Khan, Z.A., Ullah, A., Ul Haq, I., Hamdy, M., Mauro, G.M., Muhammad, K., Hijji, M., Baik, S.W.: Efficient short-term electricity load forecasting for effective energy management. Energy Reports **8**, 2373–2385 (2022)
8. Lambert, N., Gigu'ere, E., Menczel, P., Li, B., Hopf, P., Su'arez, G., Gali, M., Lishman, J., Gadhvi, R., Agarwal, R., Galicia, A., Shammah, N., Nation, P., Johansson, J.R., Ahmed, S., Cross, S., Pitchford, A., Nori, F.: Qutip 5: The quantum toolbox in Python. Physics Reports **1153**, 1–62 (2026)

9. LLC, P.I.: PJM hourly energy consumption (east region) dataset. Dataset on Kaggle (224)
10. Lopez-Piqueres, J., Chen, J., Perdomo-Ortiz, A.: Symmetric tensor networks for generative modeling and constrained combinatorial optimization. Machine Learning: Science and Technology **4**, 035009 (2023)
11. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control systems. Science **197**(4300) (1977)
12. Manzano, D.: A short introduction to the Lindblad master equation. AIP Advances **10**(2) (2020)
13. Martínez-Peña, R.: Dynamical phase transitions in quantum reservoir computing. Physical Review Letters **127**(10) (2021)
14. Metz, F., Bukov, M.: Self-correcting quantum many-body control using reinforcement learning with tensor networks. Nature Machine Intelligence **5**(7), 780–791 (2023)
15. Miller, J., Rabusseau, G., Terilla, J.: Tensor networks for probabilistic sequence modeling. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS). Proceedings of Machine Learning Research, vol. 130, pp. 3079–3087 (2021)
16. Mujal, P., Martínez-Peña, R., Nokkala, J., García-Beni, J., Giorgi, G.L., Soriano, M.C., Zambrini, R.: Opportunities in quantum reservoir computing and extreme learning machines. Advanced Quantum Technologies **4**(8) (2021)
17. Novikov, A., Podoprikhin, D., Osokin, A., Vetrov, D.P.: Tensorizing neural networks. In: Advances in Neural Information Processing Systems. vol. 28 (2015)
18. Ortín, S., Soriano, M.C., Pesquera, L., Brunner, D., San-Martín, D., Fischer, I., Mirasso, C.R., Gutiérrez, J.M.: A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron. Scientific Reports **5**(1), 14945 (Oct 2015)
19. Orús, R.: Tensor networks for complex quantum systems. Nature Reviews Physics **1**, 538–550 (2019)
20. Ran, S.J., Su, G.: Tensor networks for interpretable and efficient quantum-inspired machine learning. Intelligent Computing **2**, 0061 (2023)
21. Sannia, A., Martínez-Peña, R., Soriano, M.C., Giorgi, G.L., Zambrini, R.: Dissipation as a resource for quantum reservoir computing. Quantum **8** (2024)
22. Sannia, A., Giorgi, G.L., Zambrini, R.: Exponential concentration and symmetries in quantum reservoir computing (2025)
23. Stoudenmire, E.M.: Learning relevant features of data with multi-scale tensor networks. Quantum Science and Technology **3**(3), 034003 (2018)
24. Stoudenmire, E.M., Schwab, D.J.: Supervised learning with tensor networks. In: Advances in Neural Information Processing Systems. vol. 29 (2016)
25. Valverde, G., García-Olaizola, I., Scarpa, G., Pozas-Kerstjens, A.: Quantum-inspired tensor networks in machine learning models (2026)
26. Wang, J., Roberts, C., Vidal, G., Leichenauer, S.: Anomaly detection with tensor networks. arXiv preprint arXiv:2006.02516 (2020)
27. Yildiz, I.B., Jaeger, H., Kiebel, S.J.: Re-visiting the echo state property. Neural Networks **35** (2012)