Reservoir Computing as a Promising Approach for False Data Injection Attack Detection in Smart Grids

Carl-Hendrik Peters, and Mary Sánchez-Gordón

Faculty of Computer Science, Engineering and Economics,
Østfold University College, Halden, Norway
{carlhenp, mary.sanchez-gordon}@hiof.no

Abstract. Smart grids are highly digitalized electricity networks that are increasingly at risk from False Data Injection Attacks (FDIAs), which threaten grid stability. Although traditional machine learning techniques are well-established for detecting such anomalies, a neuromorphic computing approach like Reservoir Computing (RC) offers a promising alternative. Therefore, this study aims to explore RC for FDIA detection in Smart grids by conducting a Systematic Mapping Study to identify the current research trends and a Benchmark Evaluation of seven models across 21 simulated attack scenarios. The evaluation included the following three metrics: accuracy, robustness, and training-time efficiency. Findings show that the two traditional approaches included in this evaluation lead with up to 99 % accuracy and minimal training time. Among the five RC approaches, a Delayed Feedback Reservoir with Latency Encoding and Multi-Layer Perceptrons readout achieved ~93 % accuracy (albeit with longer training), while pairing that reservoir with a Logistic Regression readout delivered ~86 % accuracy in under 0.02 s. These findings suggest that appropriately coded and read-out RC models can serve as robust, resource-efficient solutions for real-time FDIA detection.

Keywords: Reservoir Computing, State Vector Machine, Multi-Layer Perceptrons, False Data Injection Attacks, Smart Grids, Systematic Mapping Study, Benchmark Evaluation

1 Introduction

In recent years, deep learning models have enabled incredible progress in numerous fields of application – from image and speech recognition to autonomous systems and the analysis of complex time series [1]. However, this development encounters a substantial obstacle due to the rapidly increasing computational and energy demands of modern AI models, which could hinder further progress. According to [2] the training computer doubled approximately every 6 months since entering the Deep Learning Era around 2010 – a growth rate that far exceeds both hardware and energy efficiency. An emerging AI–energy crisis raises concerns about the long-term scalability and practicality of traditional deep learning approaches [3]. As a result, the field of Green

AI (or Sustainable AI) is emerging, which seeks to mitigate the environmental impacts of AI systems by promoting energy-efficient algorithms, responsible resource usage, and transparency in reporting computational costs [4]. The increasing reliance on high computational power in AI raises not only environmental and economic concerns but also exposes a systemic vulnerability: the paradox of using energy-intensive systems to enhance resilience and security. As AI becomes integral to critical infrastructure like Smart Grid (SG), its energy demands directly affect the stability of the systems it aims to protect.

On one hand, SGs are advanced, digitalized electricity networks that use modern communication technologies and IT infrastructures to optimize the generation, distribution, and consumption of electrical energy, enabling dynamic adaptation to changes in energy flow for improved efficiency and security [5]. Cyber-attacks —one of today's most worrying problems [6, 7]— are among the most critical vulnerabilities when it comes to such systems and can possibly lead to large-scale power outages [5]. In this context, one of the most common attacks is the False Data Injection Attack (FDIA), where false information is injected into the system, causing mismanagement of energy distribution [8].

On the other hand, Reservoir Computing (RC) is a neuromorphic computing paradigm particularly well-suited for processing time-dependent data in dynamic systems like SGs. Unlike common machine learning models, which require extensive training, RC leverages a recurrent neural network (RNN) where the so-called reservoir part is fixed and randomly connected, with only the output weights being trained [9]. This architecture makes the system computationally efficient, allowing it to capture temporal patterns in data without the high computational cost associated with conventional learning methods. RC aligns with emerging research directions, highlighting edge computing and lightweight ML models that are important technologies for enabling efficient and scalable solutions in resource-constrained environments [10]. Due to these advantages, RC is becoming increasingly popular in applications such as anomaly detection, forecasting, and pattern recognition [11]. In particular, RC seems to be a viable approach for detecting dynamic attacks [12, 13].

Despite many studies carried out on FDIA detection in SGs, the use of RC remains underexplored, with only a limited number of studies addressing it [10, 13]. To fill this gap, this study aims to explore RC as a lightweight and energy-efficient ML model for FDIA detection in SGs.

2 Methods

Based on the main research objective, this exploratory study conducted a Systematic Mapping Study (SMS) to (O1) identify and analyze current research trends in the application of Reservoir Computing within the context of Smart Grids and Cybersecurity, and (O2) examine the emerging opportunities and challenges of employing Reservoir Computing for FDIA detection in Smart Grids. Then, a Benchmark Evaluation (BE) was conducted to (O3) evaluate the accuracy and

robustness of Reservoir Computing compared to traditional machine learning approaches such as Support Vector Machines and Multi-Layer Perceptrons.

These two approaches complement each other: the SMS highlights trends, challenges and opportunities in the current literature, while the BE provides empirical understandings into RCs capabilities. Moreover, a replication package for this study is available online [14].

2.1 Systematic Mapping Study

The SMS follows the guidelines proposed by Petersen et al. [15]. Initially, Hamedani's paper [12] was used as the starting point for the search. This study was identified while browsing common search engines for "Reservoir Computing", "Cybersecurity", as well as "Reservoir Computing" and "Sustainability." Next, looking through the list of references, the first backwards snowball sampling was conducted. The identified papers were then used to derive keywords for a systematic search. Synonymous terms and Boolean operators were applied to refine the results. Consequently, three key domains were defined through the following keyword substrings: 1) "Reservoir Computing" OR "Neuromorphic Computing" OR "Neuromorphic Device" OR "Liquid State Machine" OR "Echo State Network"; 2) "Smart Grid" OR "Electricity Grid" OR "Power System"; and 3) "Cybersecurity" OR "False Data Injection" OR "Attack Detection" OR "Intrusion Detection". The final search string combined these three substrings using the AND operator. **Table 1** shows the inclusion and exclusion criteria used.

Table 1. Inclusion and exclusion criteria

	Inclusion	**Exclusion**
1	Studies focus on the topic of RC in the context of cybersecurity or SGs.	Studies not presented in English or German
2	Studies focus on anomaly detection or attack detection, ideally in the context of SGs.	Books and grey literature
3	Studies are in computer science, AI or ML.	Duplicate/not online available studies

In this SMS, four databases were searched: ACM, IEEE Xplore, Springer Nature Link, and Google Scholar. These are well-known databases that retrieved 995 papers. **Fig. 1** illustrates the SMS's steps along with the number of papers in each step.

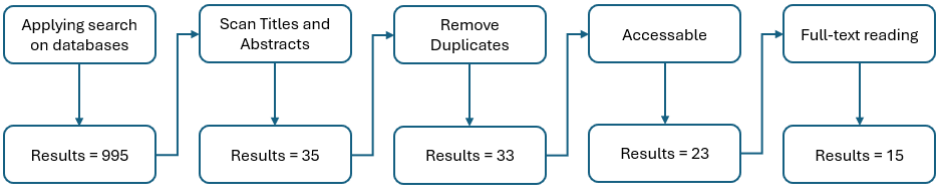

Fig. 1. Keyword-Based Search results adapted from [15]

Thirty-five studies were initially included in the SMS based on their titles, keywords, abstracts, and, when necessary, conclusions. Subsequently, two duplicate studies and ten that were not accessible online were excluded. As a result, 23 papers were selected for full-text reading. After applying the inclusion and exclusion criteria, 15 papers were finally included as primary studies (see the appendix section). These studies were then used to conduct backwards snowball sampling once again, and the same inclusion process was applied. However, additional studies have not been found to match the required inclusion criteria.

Limitations and Threats to Validity. Despite the limitations, this SMS provides a good overview of the current research landscape at the point of intersection of RC, SGs, and cybersecurity. One main limitation is the selection of databases. Although major repositories were included, it may still lead to a partial representation of the full literature. Another limitation is that the keyword-based search strategy might have excluded relevant studies that use alternative terminologies or domain-specific phrasing. Furthermore, the first inclusion and exclusion decisions were based on subjective assessments of titles and abstracts, which introduces the possibility of reviewer bias. Finally, it must be acknowledged that more literature might be available in languages other than German and English.

2.2 Benchmark Evaluation

The BE compares RC with traditional ML approaches for FDIA detection. In total, seven architectures were implemented. The traditional models were used as baselines due to their established performance and interpretability in high-dimensional classification tasks.

Experimental Setup. Similarly to Hamedani et al. [16] our benchmark evaluation used simulated data from the IEEE 57-bus test system, emulated in MATLAB using the MATPOWER open-source platform. MATPOWER was selected as a simulation platform due to its widespread adoption in previous research, availability of established benchmarks, and flexible network topology [17]. This platform offers precise control over network parameters and manipulations, which is particularly essential for reproducible evaluations of attack detection systems.

The IEEE 57-bus test system has also been used in previous studies, e.g. [16]. In our study, the system comprises 57 buses and 80 branch connections, resulting in 137 measurement points. These include both power consumption at the buses (bus_Pd) and power on the branches (branch_Pf), recorded in a structured format using the so-called meter_map.

All experiments were conducted on a local machine equipped with an AMD Ryzen 5 3500X CPU (6 cores, ~3.6GHz), 32 GB DDR4 RAM, and running Windows 10 Pro 64-Bit-Version. No GPU acceleration was used.

Datasets. To generate the data, 10,000 clean samples were first created by simulating load flows under realistic grid conditions. These functioned as a base for the subsequent

generation of targeted FDIAs, which were synthetically embedded in three attack intensity levels —weak, medium, and strong— corresponding to standard deviations (σ) of 0.02, 0.05, and 0.1, respectively. In each case, a fixed number of randomly selected measurements (20, 40, 60, 80, 100, 120 and 137) were manipulated using additive Gaussian noise. As a result, 21 compromised datasets were simulated (see the appendix in the replication package [14] for details on all 21 compromised datasets).

This noise had a mean of 0 and varying standard deviations, representing the attack intensity. This FDIA strategy simulates a stealthy injection of malicious measurements into the system's sensor data. Formally, the attacked input vector x' is given by:

$$x' = x+\varepsilon, \text{ where } \varepsilon \sim N(O, \sigma^2) \quad (1)$$

This enables analysis of the effect of different attack variables on the detection capabilities of the models while ensuring that the attack remains statistically inconspicuous yet challenging to detect, especially under lower noise settings. All attacks were constructed such that the manipulated measurements bypass basic residual-based or threshold-based anomaly detectors, making the detection task non-trivial for machine learning models. By controlling both the number of compromised variables and the noise intensity, 21 datasets were simulated, e.g., *FDI_weak_20m.mat*. They enable systematic robustness testing across a wide spectrum of attack scenarios.

To map the static input vectors into a time-dependent domain suitable for RC models, two biologically inspired encoding methods were employed:

- Latency Encoding (LE): Converts the amplitude of input features into spike timings, where higher values fire earlier. This mimics first-spike neural coding.
- Inter-Spike Interval Encoding (ISI): Converts each feature into a series of spike timings with variable temporal gaps, modulated by the feature's value.

Both methods introduce a pseudo-temporal structure to the input space, enabling temporal models (such as DFRs or spiking ESNs) to exploit temporal dynamics for enhanced detection performance.

Traditional Models. Previous studies like [18] and [13] identified State Vector Estimation (SVE) and Multi-Layer Perceptrons (MLPs) as standard supervised learning methods in the field of cybersecurity and anomaly detection in SGs. Consequently, our study includes both models.

State Vector Machine (SVM). The SVM was used with a radial basis function (RBF) kernel, which is known for its effectiveness in non-linear classification tasks [12, 13, 19]. Then, grid search was applied to optimize hyperparameters $C \in \{0.1,1,10\}$ and $\gamma \in \{\text{'scale'},0.01,0.1\}$. SVM served as a strong baseline for the detection of FDIAs on clean input features.

Multi-Layer Perceptron (MLP). A feedforward neural network was employed using one or two hidden layers with 32 to 64 neurons. Activation functions included ReLU and tanh. Hyperparameter tuning was performed via 5-fold cross-validation on a defined grid, which can be seen in the annex. This model functions as a common deep

learning baseline and is used both independently and as a readout layer in reservoir computing settings.

Reservoir Computing. ESNs and hybrid DFR models combined with either simple Logistic Regression (LR) or a MLP as the readout layer were included. For the hybrid models, the data was encoded both with LE and ISI encoding.

Echo State Network (ESN). The ESN implementation used the reservoirpy library and consisted of 700 units, a spectral radius of 1.1, and input scaling of 0.4. Ridge regression with λ=10−6 was used as the readout layer. Input data was reshaped into sequences of shape (1,137), allowing the model to process each sample as a temporal vector. Only the readout weights were trained, thereby preserving the RC paradigm.

Delayed Feedback Reservoir (DFR). The DFR model was custom-implemented following the example of [16] and operates with a single nonlinear node and a delayed feedback loop. It processes temporally encoded inputs through two variants:

- Latency Encoding (LE): Here, input values were transformed into spike times. Higher values produce earlier spikes. The encoded data is of shape (50,137), and the DFR ran with 50 reservoir units and a delay of 40 steps.
- Inter-Spike Interval Encoding (ISI): Each feature produces a spike train of fixed count (n=4), where the interval between spikes is determined by the signal strength. Input was encoded into (100,137), and the reservoir consisted of 100 units with PSP-mode – a biologically inspired input processing mechanism that simulates the temporal integration of incoming spikes by neurons – activated to mimic post-synaptic potential decay over time.

For both encoding strategies, the reservoir states were averaged over the last five timesteps. These states were then used as input for either Logistic Regression (LogReg) —A linear classifier using L2-regularization and a maximum of 1000 iterations— or MLP Readout —A neural network as described above, applied on the extracted reservoir states.

Model Implementation. All models were implemented in Python 3.9.12 using the following three frameworks: 1) scikit-learn for SVM, MLP, and LogReg; 2) reservoirpy for Echo State Networks; and 3) custom Python code for the DFR model, which can handle LE and ISI encoded data with PSP integration.

A train/test split of 80/20 was used for all models, whereby it was ensured that the same train/test split was selected for all models per data set to ensure comparability. The datasets were standardized using z-normalization and subsequently transformed into sequential formats for the RC models. Z-normalization ensures that all input features are equally weighted and can be further processed in a numerically stable manner —both by classical ML models (e.g. SVM, MLP) and by dynamically sensitive RC models [20]. The formula of the z-normalization can be formulated as:

This increases the model quality, reduces the training time and increases the reliability of the detection process. The classical ML models and the readouts of the RC models were optimized for their hyperparameter space using GridSearchCV with 5-

fold cross-validation. The best configurations were automatically saved and documented and can be found under rc_all_models_final_results in [14].

Performance Evaluation Metrics. The performance of both the RC and traditional machine learning models was evaluated using the following metrics:

Accuracy. Accuracy is a key metric for evaluating FDIA detection systems, as it quantifies their ability to correctly distinguish between attacked and secure states. It is defined as the ratio of correct predictions—true positives (TP) and true negatives (TN)—to the total number of samples, including false positives (FP) and false negatives (FN), i.e., (TP + TN) / (TP + TN + FP + FN).

Robustness. Robustness reflects the stability of a model's performance under varying conditions, such as changes in attack intensity levels or the introduction of noise in the data. This metric is critical for FDIAs detection, as attack scenarios can vary significantly in scale and complexity.

Computational Efficiency (Speed). Computational efficiency in this study was assessed by the wall-clock time of each model required to complete training. Using Python's built-in time module, the interval from the start of the fit () call to its completion was captured. These training times were recorded for every model–dataset pairing and incorporated into the results.

Limitations and Threats to Validity. Benchmarking evaluation has also several limitations. A main limitation is the synthetically generated datasets, since they lack irregularities, uncertainties, and adversarial behaviors typically represented in real-world data. Consequently, the models' performances may be different in real-world scenarios. However, it provides preliminary insights into the overall capabilities, robustness, and comparative strengths of different detection models under controlled and repeatable conditions, which lays the foundation for future evaluations on real-world applications.

Another limitation is that time was used as a proxy for energy efficiency. Future work should be conducted on cloud infrastructure or energy-monitored environments to enable accurate sustainability analysis. It would be interesting to use CarbonTracker to measure carbon footprint and energy consumption. It was not feasible in our study due to technical constraints. Our experiments ran on a local CPU-only machine without access to GPUs or high-performance clusters. CarbonTracker is designed to work best with server-grade or cloud-based environments, where detailed power drawing can be captured from hardware interfaces such as NVIDIA's NVML or server power sensors. Moreover, the used workstation did not expose necessary energy consumption data via system APIs, which CarbonTracker relies on for accurate estimates.

Additionally, the study focuses only on a controlled type of cyberattack, called additive FDIAs with Gaussian noise, applied in varying intensities and scope. While this design facilitates systematic comparisons, it does not account for more advanced attack types such as coordinated attacks, stealthy FDIAs, replay, or data integrity

violations. Hence, the robustness conclusions are limited to basic FDIA scenarios. Also, all experiments are conducted on the IEEE 57-bus system. While it represents a moderately complex topology suitable for benchmarking, it may not capture the variability found in larger or more dynamically changing grids.

Furthermore, all models were trained and tested using the same 80/20 data split for each dataset to ensure consistency. While this improves comparability, it also introduces potential for variance due to sampling bias. Although cross-validation was used during hyperparameter tuning, the final model evaluations could benefit from repeated random splits or k-fold validation to improve statistical robustness.

Finally, the implemented hybrid models involve temporal encodings and dynamic reservoir behavior, making them harder to interpret compared to the classic ML models. This complexity may hinder real-world acceptance and deployment.

3 Results

The results are presented according to the answers to the research questions posed in this study.

3.1 Research Trends (O1) including Opportunities and Challenges (O2)

The SMS provides a structured overview of the current state of the art of this topic. RC is an emerging yet underexplored approach in SG cybersecurity. While various architectures and encoding schemes have been proposed, these are often developed and tested under controlled, simulated environments.

Fig. 2 shows the techniques in terms of research type and performance metric used in primary studies. Here, it can be observed that experimental studies dominate, while surveys, conceptual work, and hardware-oriented investigations are less common. The most frequently addressed performance metrics are accuracy, robustness, and training time, reflecting the priority of practice-relevant evaluation criteria. Models such as MLPs, SVMs, and DFRs are particularly prominent in terms of both research activity and performance measurement, underscoring their relevance for FDIA detection. Neuromorphic approaches, such as SNNs and ESNs, are mainly investigated in experimental contexts, but have lower publication volumes and more heterogeneous areas of application. There is a noticeable shift in research trends toward resource- and energy-efficient algorithms, as manifested by the attention paid to energy efficiency and scalability metrics, reflecting the current paradigm shift toward sustainable AI. In other words, there is an emerging interest in hybrid and neuromorphic reservoir computing architectures as promising alternatives for the real-time monitoring of subjectively complex and resource-constrained systems such as smart grids.

Twelve of the fifteen primary studies were published after 2019, indicating that this topic has gained traction in the past five years. A possible reason for that is the increased digitalization of power grids, and the growing cyber threat levels. Indeed, this trend is aligned with the increasing relevance of SGs and cyber-physical systems [21]. However, it must be noted that the topic is still in its infancy since the top three

institutions contributing to this area are Virginia Polytechnic Institute (5 papers), followed by the University of Chile (3), and the University of Kansas (3).

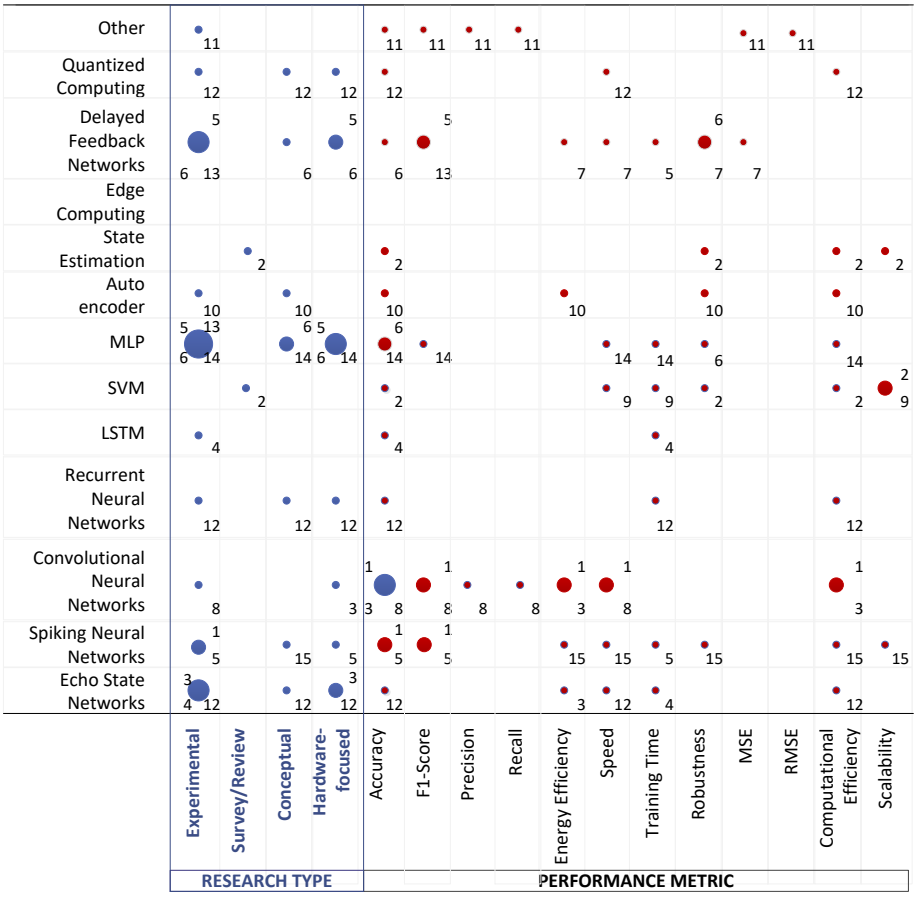

Fig. 2. Overview of traditional machine learning techniques and RC in terms of research type and performance metric. Note: Bubble size = publication count and [#] = source reference.

RC is repeatedly identified as a paradigm offering superior computational and energy efficiency, achieved by separating a fixed, randomized reservoir from a simple-to-train linear readout layer [22–24]. Due to their low computational footprint, RC models are well suited for real-time implementation on resource-constrained edge devices within SGs [24, 25]. Particularly SNNs and DFRs, demonstrate notable robustness against dynamic and stealthy FDIAs by leveraging temporal encoding to capture subtle data deviations [12, 16]. Additionally, a clear research trend is emerging towards hybrid and neuromorphic architectures, integrating RC with deep learning, federated learning, and brain-inspired hardware to enhance accuracy, decentralization, and energy efficiency [25–27]. Despite promising results, critical research gaps remain, primarily a lack of empirical validation in real-world deployments and studies on the

interoperability and systemic integration of these models into existing Smart Grid infrastructures [12, 28].

Opportunities for RC within the context of FDIA detection in SGs are vast. RC models can offer very efficient training procedures, hence be computationally extremely efficient, while displaying solid detection accuracy [22–24]. This could be of high relevance in resource constrained environments, including – amongst others – for example edge devices. Depending on the respective context, various (hybrid) models can be implemented in different (hyperparameter-related) ways, which all have their own strengths and weaknesses which need to be explored in further research.

A major challenge in this field, however, is the requirement to temporally encode the data which the models work with. Depending on the respective context, environment, and associated task, these encoding strategies can cause disproportionate efforts, subsequent interpretability difficulties, or are incompatible with certain RC architectures. Future research should examine which RC models in particular work well with which kind of encoding strategies, respective data formats, and whether different tasks also influence the resulting success of the models.

3.2 Benchmark Evaluation (O3)

The benchmark evaluation included five reservoir computing models and two traditional machine learning approaches. It provides preliminary results about the performance of the selected models for FDIA detection in SGs. The results offer some insights about accuracy, robustness, and computational efficiency under varying attack intensities and numbers of compromised meters. As expected, a trend emerged across nearly all models: detection accuracy increases with both attack amplitude and number of compromised measurements (see **Fig. 3**).

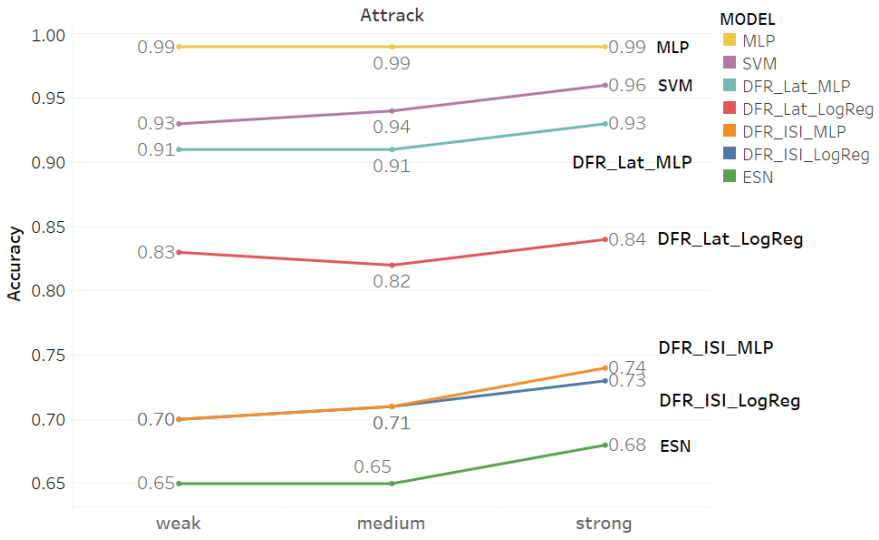

Fig. 3. Accuracy across different levels of attack strength.

This pattern is especially evident in SVM and MLP, where performance approaches or even reaches 100% under strong, wide-scope attack conditions. The explanation is intuitive: larger or more widespread anomalies produce stronger signal deviations, which are more easily detected by the models. The MLP consistently achieved the highest detection accuracy in all datasets, with a median accuracy of ~0.99. Also, it exhibited very good efficiency with the median training time being ~ 4.5 seconds. SVM also proved very effective (median accuracy ~0.97) and even a little more efficient than the MLP variant, recording a median training time of 4.2 seconds. In the RC domain, the best-performing architecture was the DFR with LE and MLP readout, reaching up to 96% accuracy. With a median training time of ~21 seconds, this hybrid architecture, however, takes a lot longer to train when compared, for example, to both traditional ML approaches.

Interestingly, the simpler variant using LR as readout also performed remarkably well (up to ~89%), while being extremely fast to train (with training times as little as 0.0164 seconds). The ESN, while conceptually simpler and faster to train than other architectures, reached only moderate accuracy (maximum ~73%). The model also displayed comparably long training time (~14.5 seconds median training time) across datasets while maintaining only moderate accuracy.

A surprising and consistent finding was the underperformance of DFR_ISI model variants, with detection performance declining as the number of compromised meters increased. Even when paired with an MLP readout, ISI-encoded DFRs failed to exceed ~73% accuracy and often fell below 70%. Moreover, they exhibited very long training times (up to 53.5202 seconds), significantly exceeding those of even the full MLP classifiers. This suggests a crucial mismatch between the model's architecture and the ISI encoding strategy. Concretely, a plausible explanation for this underperformance lies in the increased complexity and sparsity of ISI spike trains, which may make it harder for the DFR itself to reliably extract discriminative temporal features.

As expected, LR readouts trained orders of magnitude faster than MLP-based models. The DFR+Latency+LogReg model, for instance, completed training in less than 21 milliseconds across all datasets, with the longest recorded training time being 0.0209 seconds. This makes it a compelling option for time- and energy-constrained environments. However, accuracy-tradeoffs must be acknowledged and expected in such cases. By contrast, the DFR+ISI+MLP hybrid exhibited the worst trade-off, combining poor performance with high training time. These results underscore the importance of aligning encoding strategies with reservoir dynamics and readout complexity.

In summary, the traditional ML approaches —MLP and SVM— demonstrated the best overall performances in this setting, combining very high accuracy with moderate training effort, while maintaining consistent results across all attack intensities. Among the RC models, DFR_LAT variants show potential as competitive alternatives: DFR_LAT_LogReg stands out significantly for its extreme efficiency but needs improved accuracy, while DFR_LAT_MLP delivers strong accuracy but requires faster training. These trade-offs are clearly visible in **Fig. 4**. In contrast, ISI-based variants are not viable, as they express high training times and consistently poor accuracy.

Similarly, the ESN model, though being lightweight, lacks both accuracy and training efficiency to be considered a serious contender in this benchmark.

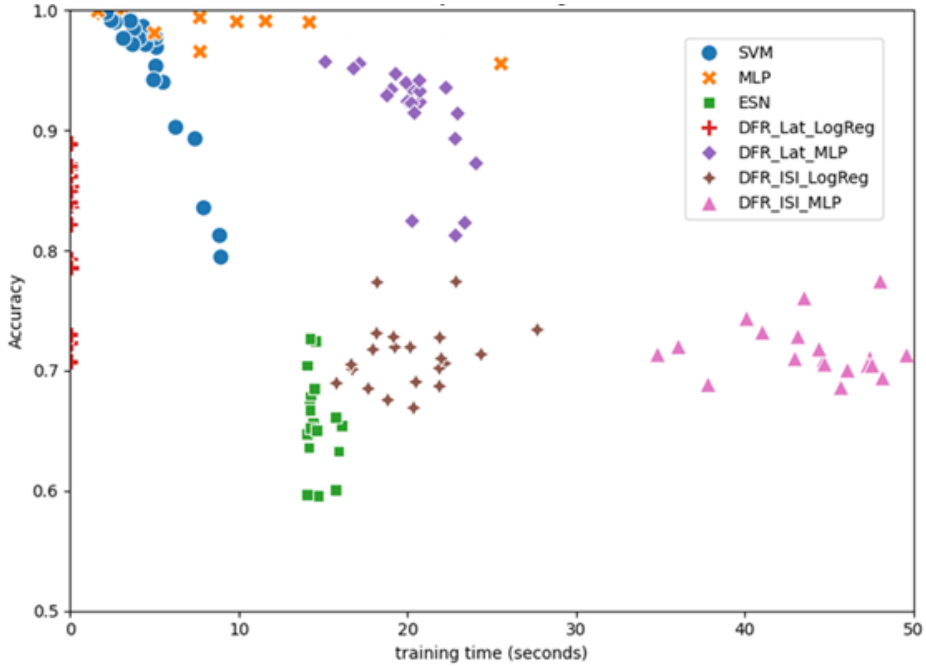

Fig. 4. Seven models' trade-offs between detection accuracy and training time.

4 Discussion

The results from this work suggest that RC-based architectures, in particular DFNs with LE and a simple readout layer like Logistic Regression, are highly promising for SGs environments. Their low training and inference complexity make them ideal for microcontroller-based or embedded systems, where traditional deep learning models like MLPs may not be feasible [12, 16, 23]. At the same time, interpretability and maintainability must be considered. While classical models like SVMs offer clearer decision boundaries, DFRs —especially those using temporal encoding— may be harder to interpret and diagnose in failure cases [18, 19]. Thus, model transparency and fault explainability remain important areas for further research and deployment readiness.

Modern AI models are increasingly criticized for their rising energy demands and environmental footprint [3]. This challenge is also present in the domain of real-time, infrastructure-level monitoring, where scalability and low energy consumption are not optional but essential [4]. While direct measurement of energy consumption (e.g., via CarbonTracker) could not be conducted in our study, training time served as a practical proxy. The results clearly show that the DFR_LAT variant requires orders of magnitude

—significantly— less time and resources than conventional ML approaches while maintaining competitive accuracy. This makes them well-suited for edge deployment scenarios, including remote substations and autonomous grid segments. Moreover, the ability to deploy AI directly at the edge has secondary sustainability benefits, including reduced bandwidth usage, increased data sovereignty, and lower reliance on cloud infrastructure, all of which contribute to more energy-conscious system design [16, 29, 30]. While the analysis delivers key insights into the potential of RC for SG cybersecurity, it acknowledges limitations related to dataset representation and the simplification of attack models.

Despite promising theoretical and empirical results, a significant lack of empirical validation in real smart grid environments remains. This research gap has been highlighted by earlier studies emphasizing the importance of evaluating models under practical conditions [12, 13, 28]. Moreover, the requirements and complexities of temporal encoding strategies present challenges that need further investigation, as they affect both training effort and model interpretability [12, 16].

The significance of performance differences among models was assessed using two non-parametric tests, Friedman and Wilcoxon. The results confirmed that all models differ significantly in terms of accuracy and training time ($p < 0.001$). MLP achieved the highest detection accuracy (~0.99), statistically significantly outperforming all other models, while also maintaining low training times. SVM followed closely, offering a robust and efficient alternative. The DFR_LAT_LogReg model demonstrated significantly better training efficiency with near-zero training time (~0.0s), making it highly suitable for constrained environments, despite its significantly lower accuracy (~0.89). DFR+Latency+MLP reached a moderate balance between strong performance in terms of accuracy (~0.93) and poor efficiency (~21s training time). ESN showed only weak performance regarding accuracy (~0.66) and training time (~15s), positioning an inviable alternative to traditional approaches. ISI-based DFR models performed the weakest, showing significantly lower accuracy (~0.70) combined with the highest training times (~44s), classifying them as inaccurate and inefficient.

These results are both statistically significant and consistent across all evaluations, underscoring the importance of aligning model complexity, encoding strategy, and computational resources in applications.

5 Conclusions

This study explores the potential of RC for detecting FDIAs in SGs. By combining an SMS with a BE, it provides both an overview of the academic landscape and a practical, performance-oriented comparison of traditional and RC-based detection models.

Overall, our study suggests that Reservoir Computing is a promising approach for scalable, sustainable, and accurate cyberattack detection in Smart Grids —particularly when combined with encoding strategies and minimalistic readouts like Logistic Regression. Its potential lies in balancing computational simplicity with temporal

processing capacity, making it a valuable alternative in the transition to intelligent, resilient, and sustainable energy systems.

Future research should address the identified limitations and research gaps by collecting real-world datasets and examining RC performances in real-world applications, applying more diverse attack types, exploring the energy footprint of models more precisely, e.g., via hardware-level power profiling and improving the interpretability and transparency of RC models to support their adoption in critical infrastructure.

6 Appendix: Primary Studies

Ref	Title	Authors	Year
1	A revolutionary approach to use convolutional spiking neural networks for robust intrusion detection	Lin Y., Xu X., Xu H.	2024
2	Brief Survey on Attack Detection Method for Cyber-Physical Systems	Tan S., Guerrero J., Xie P., Han R., Vasquez J.	2020
3	Convolutional Echo-State Network with Random Memristors for Spatiotemporal Signal Classification	Wang S., Chen H., Zhang W., Li Y., Wang D., Shi S., Zhao Y., Loong K.C., Chen X., Dong Y., Zhang Y., Jiang Y., Furqan C., Chen J., Wang Q., Xu X., Wang G., Yu H., Shang D., Wang Z.	2022
4	Cyberattack Detection in Smart Grids based on Reservoir Computing	Kim K., Sasahara H., Imura J.	2023
5	Detecting Dynamic Attacks in Smart Grids Using Reservoir Computing: A Spiking Delayed Feedback Reservoir Based Approach	Hamedani K., Liu L., Hu S., Ashdown J., Wu J., Yi Y.	2019
6	Emerging Applications of Reservoir Computing in Cyber Physical Systems Security	Hamedani K.	2018
7	Enabling Sustainable Cyber Physical Security Systems Through Neuromorphic Computing	Li J., Liu L., Zhao C., Hamedani K., Atat R., Yi Y.	2023
8	False Data Injection Attack Detection for Secure Distributed Demand Response in Smart Grids	Dayaratne T., Salehi M., Rudolph C., Liebman A.	2022
9	How Machine Learning Can Support Cyber-Attack Detection in Smart Grids	Zarpelão B. B., Barbon Jr. S., Acarali D., Rajarajan M.	2020
10	Network Intrusion Detection for Cyber Security on Neuromorphic Computing System	Alom Z., Taha T. M.	2017
11	Next Generation Automated Reservoir Computing for Cyber Defense	Demertzis K., Iliadis L.	2023
12	Quantized Reservoir Computing on Edge Devices for Communication Applications	Liu S, Lingjia L, Yi, Y.	2020
13	Reservoir Computing Meets Smart Grids: Attack Detection Using Delayed Feedback Networks	Hamedani K., Liu L., Atat R., Wu J., Yi Y.	2018
14	The Novel Applications of Deep Reservoir Computing in Cyber-Security and Wireless Communication	Hamedani K., Zhou Z., Bai K., Liu L.	2020
15	Towards a Federated Intrusion Detection System based on Neuromorphic Computing	Lofù D., Sorino P., Di Noia T., Di Sciascio E.	2024

References

1. Maslej, N., Fattorini, L., Perrault, R., Parli, V. Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Niebles, J.C., Shoham, Y., Wald, R., Clark, J.: Artificial Intelligence Index Report 2024. (2024). https://doi.org/10.48550/arXiv.2405.19522.
2. Heim, L., Koessler, L.: Training Compute Thresholds: Features and Functions in AI Regulation, http://arxiv.org/abs/2405.10799, (2024). https://doi.org/10.48550/arXiv.2405.10799.
3. Lannelongue, L., Grealey, J., Inouye, M.: Green Algorithms: Quantifying the Carbon Footprint of Computation. Advanced Science. 8, 2100707 (2021). https://doi.org/10.1002/advs.202100707.
4. Bolón-Canedo, V., Morán-Fernández, L., Cancela, B., Alonso-Betanzos, A.: A review of green artificial intelligence: Towards a more sustainable future. Neurocomputing. 599, 128096 (2024). https://doi.org/10.1016/j.neucom.2024.128096.
5. Moreno Escobar, J.J., Morales Matamoros, O., Tejeida Padilla, R., Lina Reyes, I., Quintana Espinosa, H.: A Comprehensive Review on Smart Grids: Challenges and Opportunities. Sensors. 21, 6978 (2021). https://doi.org/10.3390/s21216978.
6. Griffiths, C.: The Latest Cyber Crime Statistics (updated October 2024) | AAG IT Support, https://aag-it.com/the-latest-cyber-crime-statistics/, last accessed 2024/11/15.
7. World Economic Forum: Global Risks Report 2024, https://www.weforum.org/publications/global-risks-report-2024/, last accessed 2024/11/15.
8. Gunduz, M.Z., Das, R.: Cyber-security on smart grid: Threats and potential solutions. Computer Networks. 169, 107094 (2020). https://doi.org/10.1016/j.comnet.2019.107094.
9. Schrauwen, B., Verstraeten, D., Van Campenhout, J.: An overview of reservoir computing: theory, applications and implementations. (2007).
10. Reda, H.T., Anwar, A., Mahmood, A.: Comprehensive Survey and Taxonomies of False Injection Attacks in Smart Grid: Attack Models, Targets, and Impacts. Renewable and Sustainable Energy Reviews. 163, 112423 (2022). https://doi.org/10.1016/j.rser.2022.112423.
11. Tanaka, G., Yamane, T., Héroux, J.B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., Hirose, A.: Recent Advances in Physical Reservoir Computing: A Review. Neural Networks. 115, 100–123 (2019). https://doi.org/10.1016/j.neunet.2019.03.005.
12. Hamedani, K.: Emerging Applications of Reservoir Computing in Cyber Physical Systems Security. (2018).
13. Meydani, A., Shahinzadeh, H., Ramezani, A., Nafisi, H., Gharehpetian, G.B.: A Review and Analysis of Attack and Countermeasure Approaches for Enhancing Smart Grid Cybersecurity. In: 2024 28th International Electrical Power Distribution Conference (EPDC). pp. 1–19. IEEE, Zanjan, Iran, Islamic Republic of (2024). https://doi.org/10.1109/EPDC62178.2024.10571761.
14. Peters, C-H., Sánchez-Gordón, M, (2025). Online Resource: Reservoir Computing as a Promising Approach for False Data Injection Attack Detection in Smart Grids. figshare. https://doi.org/10.6084/m9.figshare.30141511.
15. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology. 64, 1–18 (2015). https://doi.org/10.1016/j.infsof.2015.03.007.
16. Hamedani, K., Liu, L., Hu, S., Ashdown, J., Wu, J., Yi, Y.: Detecting Dynamic Attacks in Smart Grids Using Reservoir Computing: A Spiking Delayed Feedback Reservoir Based Approach. IEEE Trans. Emerg. Top. Comput. Intell. 4, 253–264 (2020). https://doi.org/10.1109/TETCI.2019.2902845.

17. Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J.: MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education. IEEE Transactions on Power Systems. 26, 12–19 (2011). https://doi.org/10.1109/TPWRS.2010.2051168.
18. Dayaratne, T., Salehi, M., Rudolph, C., Liebman, A.: False Data Injection Attack Detection for Secure Distributed Demand Response in Smart Grids. In: 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 367–380. IEEE, Baltimore, MD, USA (2022). https://doi.org/10.1109/DSN53405.2022.00045.
19. Ozay, M., Esnaola, I., Yarman Vural, F.T., Kulkarni, S.R., Poor, H.V.: Machine Learning Methods for Attack Detection in the Smart Grid. IEEE Trans Neural Netw Learn Syst. 27, 1773–1786 (2016). https://doi.org/10.1109/TNNLS.2015.2404803.
20. Montoya, J.: Lecture 02: Data Handling. Introduction to Artificial Intelligence. , Hochschule Luzern: Departement für Informatik (2023).
21. Dimensions.ai: Overview for Reservoir Computing and Smart Grid and Cybersecurity, https://app.dimensions.ai/analytics/publication/overview/timeline?search_mode=content&search_text=Reservoir%20Computing%20and%20Smart%20Grid%20and%20Cybersecurity&search_type=kws&search_field=full, last accessed 2025/06/11.
22. Hamedani, K., Liu, L., Atat, R., Wu, J., Yi, Y.: Reservoir Computing Meets Smart Grids: Attack Detection Using Delayed Feedback Networks. IEEE Trans. Ind. Inf. 14, 734–743 (2018). https://doi.org/10.1109/TII.2017.2769106.
23. Kim, K., Sasahara, H., Imura, J.: Cyberattack Detection in Smart Grids based on Reservoir Computing. IFAC-PapersOnLine. 56, 971–976 (2023). https://doi.org/10.1016/j.ifacol.2023.10.1691.
24. Liu, S., Liu, L., Yi, Y.: Quantized Reservoir Computing on Edge Devices for Communication Applications. In: 2020 IEEE/ACM Symposium on Edge Computing (SEC). pp. 445–449. IEEE, San Jose, CA, USA (2020). https://doi.org/10.1109/SEC50012.2020.00068.
25. Alom, M.Z., Taha, T.M.: Network intrusion detection for cyber security on neuromorphic computing system. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 3830–3837. IEEE, Anchorage, AK, USA (2017). https://doi.org/10.1109/IJCNN.2017.7966339.
26. Li, J., Liu, L., Zhao, C., Hamedani, K., Atat, R., Yi, Y.: Enabling Sustainable Cyber Physical Security Systems through Neuromorphic Computing. IEEE Trans. Sustain. Comput. 3, 112–125 (2018). https://doi.org/10.1109/TSUSC.2017.2717807.
27. Lofù, D., Sorino, P., Di Noia, T., Di Sciascio, E.: Towards a Federated Intrusion Detection System based on Neuromorphic Computing. In: 2024 9th International Conference on Smart and Sustainable Technologies (SpliTech). pp. 1–5. IEEE, Bol and Split, Croatia (2024). https://doi.org/10.23919/SpliTech61897.2024.10612534.
28. Zarpelão, B.B., Barbon Jr., S., Acarali, D., Rajarajan, M.: How Machine Learning Can Support Cyberattack Detection in Smart Grids. In: Artificial Intelligence Techniques for a Scalable Energy Transition: Advanced Methods, Digital Technologies, Decision Support Tools, and Applications. pp. 225–259. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-42726-9.
29. Cao, K., Liu, Y., Meng, G., Sun, Q.: An Overview on Edge Computing Research. IEEE Access. 8, 85714–85728 (2020). https://doi.org/10.1109/ACCESS.2020.2991734.
30. Zhang, W., Yao, P., Gao, B., Liu, Q., Wu, D., Zhang, Q., Li, Y., Qin, Q., Li, J., Zhu, Z., Cai, Y., Wu, D., Tang, J., Qian, H., Wang, Y., Wu, H.: Edge learning using a fully integrated neuro-inspired memristor chip. Science. 381, 1205–1211 (2023). https://doi.org/10.1126/science.ade3483.