Forward–Forward Learning for Imbalanced Tabular Predictive Maintenance on a Real-World Smart-Grid Fault Dataset

Enrico De Santis*, Danial Zendehdel*, Gianluca Ferro*, and Antonello Rizzi*
*Department of Information Engineering, Electronics and Telecommunications
University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy
Email: {enrico.desantis, danial.zendehdel, gianluca.ferro, antonello.rizzi}@uniroma1.it

***Abstract*—Forward–Forward learning replaces global error back-propagation with a layer-local objective based on the goodness of forward activations. This paper evaluates a stabilized, Trifecta-inspired Forward–Forward formulation for imbalanced tabular predictive maintenance on a real-world smart-grid fault dataset. A supervised label-embedding Forward–Forward network is assessed under stratified i.i.d. train/test splits, with cross-validation on the training portion and final reporting on held-out test sets. Comparative experiments against standard tabular baselines show that Forward–Forward is competitive under class imbalance and remains close to back-propagation multi-layer perceptrons, while random forests provide the strongest overall performance. Probability calibration is also examined to support risk-aware decision making, and goodness-margin dynamics together with single-factor ablations highlight the role of LayerNorm as a key stabilizer for reliable training. Timing measurements quantify training and inference cost.**

***Index Terms*—back-propagation, fault detection, forward–forward learning, imbalanced classification, predictive maintenance, smart grids, tabular learning**

I. Introduction

Predictive maintenance for medium-voltage power grids requires early identification of fault conditions from heterogeneous measurements, often under pronounced class imbalance. In operational settings, probability-like fault scores support risk-aware inspection scheduling and prioritization. Modern machine learning approaches have improved fault detection performance, while gradient-based back-propagation remains the dominant training paradigm for multi-layer networks [1].

Forward–Forward learning was introduced by Geoffrey Hinton as a biologically inspired alternative to error back-propagation (BP) [2]. The proposal is motivated by the long-standing observation that standard BP relies on non-local information and global gradient transport, whereas biological learning is commonly modeled through local synaptic updates driven by forward activity and by complementary mechanisms that suppress spurious internal configurations. In this perspective, early theoretical accounts of sleep and dreaming emphasize a functional role in weakening parasitic or hallucination-like modes, suggesting the need for a "negative" learning phase that counterbalances reinforcement on real data [3].

Forward–Forward operationalizes this idea in a purely forward framework, sculpting the training loss locally at each layer (i.e., optimizing a local objective) through the contrast between positive and negative samples, without propagating error gradients across the full network [2]. In other words, this learning paradigm replaces global error back-propagation with independent layer-wise optimization driven by forward activations.

Intuitively, the layer "goodness" measures how strongly the layer responds to an input, typically via the average squared activation magnitude. During training, goodness is encouraged to be high for positive (correctly labeled) inputs and low for negative (incorrectly labeled) inputs, so that the network learns internal representations that separate compatible from incompatible label–input pairs. Because this objective directly depends on activation magnitudes, normalization plays a special role in stabilizing Forward–Forward learning. Techniques such as LayerNorm prevent trivial scale growth and reduce sensitivity to input scale, enabling deeper and more reliable training dynamics [4], [5].

This paper investigates whether a stabilized, Trifecta-inspired Forward–Forward formulation is compatible with imbalanced tabular predictive maintenance and can achieve competitive performance relative to strong tabular baselines under a standard stratified evaluation protocol. A supervised Forward–Forward model with label embedding is implemented and evaluated on a real-world smart-grid fault dataset across multiple by-seed splits, with cross-validation on the training portion and final reporting on held-out test sets. The real-world dataset is derived from the medium-voltage (MV) power distribution grid of the city of Rome, operated by Azienda Comunale Energia e Ambiente (ACEA), and was constructed and validated together with experienced field engineers from the utility. The experiments indicate that Forward–Forward attains strong PR-AUC and F1 under class imbalance and remains close to back-propagation MLP baselines, while random forests provide the strongest overall performance. Additional analyses based on goodness-margin dynamics and single-factor ablations further support the interpretation that LayerNorm is a key stabilizer for reliable learning in this setting.

This study is organized as follows. Section II reviews related work. Section III describes the Forward–Forward methodology adopted in this study. Section IV details the dataset and evaluation protocol. Section V reports the experimental results

and analysis. Section VI concludes the paper.

II. Related Work

Back-propagation formalizes learning in differentiable multi-layer networks through error gradients propagated across layers [1]. Forward–Forward learning instead trains each layer through a local objective that contrasts positive and negative samples using only forward computations [2]. Several extensions have introduced predictive variants and stabilization techniques [5], [6], and applications to convolutional architectures have been explored [7].

Fault prediction and recognition in power distribution networks have been extensively studied using supervised learning on tabular utility data, combining static asset descriptors with time-varying operational and environmental covariates. Early large-scale deployments leveraged such features to rank feeders by near-term failure susceptibility and support preventive maintenance [8], while subsequent work extended this approach into decision-support frameworks integrating heterogeneous historical and real-time data [9]. More recent studies explicitly model the interaction between exogenous stressors and endogenous operating conditions, for instance under extreme-weather regimes, using engineered features from meteorological and load data [10].

Beyond vision and generic tabular benchmarks, forward–forward learning has been explored in a small number of studies on fault detection and isolation for multivariate industrial process monitoring. A variance-capturing forward–forward autoencoder (VFFAE) combines layer-wise FF training with an autoencoder structure to perform unsupervised fault detection on classical industrial benchmarks such as the Tennessee–Eastman process and wastewater treatment datasets [11]. Related work on forward-propagating neural networks has focused on model maintenance and real-time updating under streaming conditions rather than supervised fault classification [12]. To the best of our knowledge, no prior work has applied forward–forward learning to fault detection or predictive maintenance in medium-voltage power distribution networks or smart grid settings.

III. Methods

A. Problem Formulation

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ denote a labeled dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is a tabular feature vector and $y_i \in \{0, 1\}$ denotes the fault label. The learning task is binary classification under class imbalance, with the goal of producing discriminative scores for fault ranking and decision-making.

B. Label Embedding for Supervised Forward–Forward

Forward–Forward learning can be specialized to supervised classification by augmenting inputs with a label embedding. Let $\mathbf{e}_y \in \mathbb{R}^2$ denote a one-hot encoding of y. The augmented representation is defined as:

$$\tilde{\mathbf{x}} = [\mathbf{x}; \mathbf{e}_y] \in \mathbb{R}^{d+2}. \tag{1}$$

C. Layerwise Goodness and Trifecta-Inspired Objective

Consider a feed-forward network composed of L layers. For layer ℓ, let $\mathbf{h}_\ell \in \mathbb{R}^{d_\ell}$ denote its activation, computed as $\mathbf{h}_\ell = f_\ell(\mathbf{h}_{\ell-1})$, where $\mathbf{h}_0 = \tilde{\mathbf{x}}$. A scalar goodness for layer ℓ is defined as:

$$g_\ell(\mathbf{h}_{\ell-1}) = \frac{1}{d_\ell}\|\mathbf{h}_\ell\|_2^2. \tag{2}$$

The total goodness is defined as $G(\tilde{\mathbf{x}}) = \sum_{\ell=1}^{L} g_\ell(\mathbf{h}_{\ell-1})$.

For each training example $(\mathbf{x}, y)$, a positive sample is constructed as $\tilde{\mathbf{x}}^+ = [\mathbf{x}; \mathbf{e}_y]$, and a negative sample as $\tilde{\mathbf{x}}^- = [\mathbf{x}; \mathbf{e}_{1-y}]$. For layer ℓ, let $g_\ell^+ = g_\ell(\mathbf{h}_{\ell-1}^+)$ and $g_\ell^- = g_\ell(\mathbf{h}_{\ell-1}^-)$ denote the goodness computed on the forward activations induced by $\tilde{\mathbf{x}}^+$ and $\tilde{\mathbf{x}}^-$, respectively. Each layer is trained by contrasting g_ℓ^+ and g_ℓ^- with a symmetric margin controlled by a threshold $T > 0$:

$$\begin{aligned}\mathcal{L}_\ell = \mathbb{E}\left[\log\left(1 + \exp\left(T - g_\ell^+\right)\right)\right] \\ + \mathbb{E}\left[\log\left(1 + \exp\left(g_\ell^- - T\right)\right)\right].\end{aligned} \tag{3}$$

Here $\mathbb{E}[\cdot]$ denotes the empirical average over the mini-batch.

In the implementation used in this study, each layer uses a linear map W_ℓ followed by LayerNorm $\mathrm{LN}(\cdot)$ and a pointwise nonlinearity $\phi(\cdot)$ (ReLU), and optionally applies ℓ_2 normalization to its input to mitigate scale variability in tabular features. This configuration follows the Trifecta stabilizations for deeper Forward–Forward training [5]:

$$\mathbf{h}_\ell = \phi\Big(\mathrm{LN}\Big(W_\ell\, \hat{\mathbf{h}}_{\ell-1}\Big)\Big), \quad \hat{\mathbf{h}}_{\ell-1} = \frac{\mathbf{h}_{\ell-1}}{\max(\|\mathbf{h}_{\ell-1}\|_2, \varepsilon)}. \tag{4}$$

where $\varepsilon > 0$ is a small constant for numerical stability. Training follows the Forward–Forward principle. Parameters (W_ℓ, LN) are updated with a layer-local optimizer (Adam) using only $\mathcal{L}_\ell$, and activations are detached between layers so that no global error gradients are propagated. The Trifecta contribution adopted here is the symmetric loss in (3) together with activation normalization (and input normalization), which empirically improves stability and prevents goodness saturation in deeper networks; overlapping updates are not used in this implementation. A schematic summary is provided in Fig. 1.

D. Inference

Given $\mathbf{x}$, goodness scores are computed for both candidates $y \in \{0, 1\}$. A probability-like output is obtained via a softmax mapping:

$$p(y = 1 \mid \mathbf{x}) = \frac{\exp\left(G([\mathbf{x}; \mathbf{e}_1])\right)}{\exp\left(G([\mathbf{x}; \mathbf{e}_0])\right) + \exp\left(G([\mathbf{x}; \mathbf{e}_1])\right)}. \tag{5}$$

IV. Experimental Setup

A. Dataset

Experiments are conducted on a real-world fault detection dataset collected from the medium-voltage (MV) power distribution grid of the city of Rome, operated by Azienda Comunale Energia e Ambiente (ACEA). The grid consists of radial MV backbones with nominal voltage of 20 kV (with a limited number of legacy 8.4 kV lines), supplied

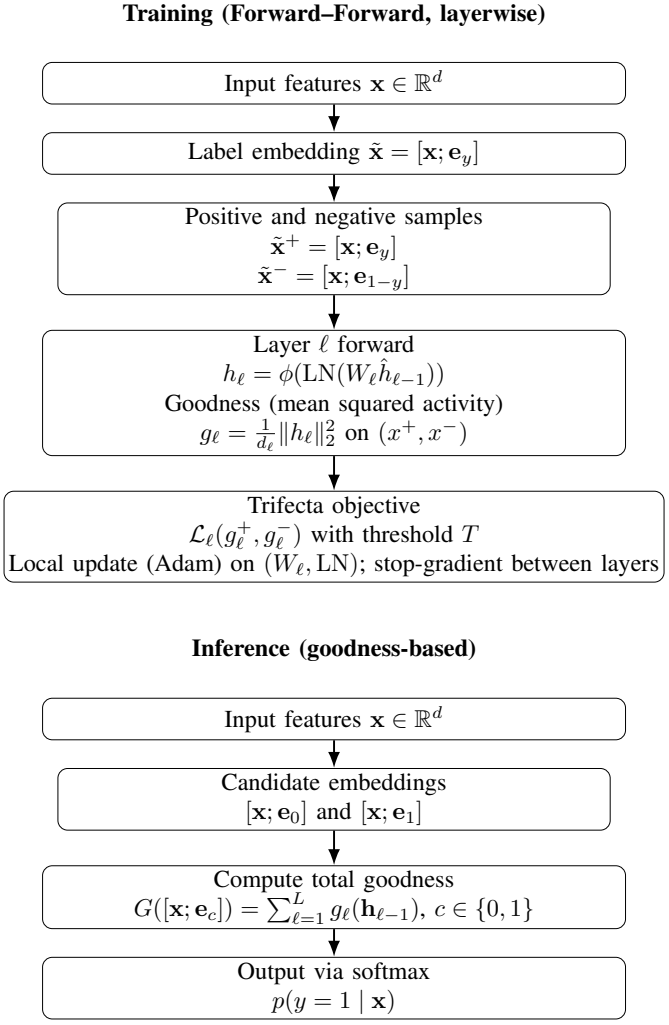

Fig. 1. Forward–Forward workflow for binary tabular fault detection, separating the training phase (top) from goodness-based inference (bottom).

by pairs of primary stations and designed to allow counter-supply in case of branch failure. The MV network extends over approximately 10,490 km and feeds a large number of secondary stations through feeders equipped with protection and switching devices to preserve radial operation during fault conditions [13].

The dataset is derived from an integrated monitoring and information infrastructure supporting smart grid operations. Data are collected from heterogeneous company systems, including supervisory control and data acquisition, alarm and protection systems, geographic and territorial information systems, and remote control platforms. Features originate from smart sensors distributed across the network as well as from quantities computed using the electrical model and operating diagrams of the grid. Feature selection and data validation were performed in collaboration with field experts, following an in-depth analysis of network failures and malfunctioning mechanisms [13].

Fault events considered in this study correspond to localized faults lasting longer than three minutes, which are operationally relevant for maintenance and diagnostic activities. After data cleaning and transformation, the resulting dataset consists of $N = 5342$ instances described by $d = 16$ tabular features, of which 751 correspond to fault conditions and 4591 to normal operation. The positive rate is approximately 0.141, resulting in a pronounced class imbalance that reflects realistic predictive maintenance scenarios [13].

TABLE I
DATASET SUMMARY AND EVALUATION PROTOCOL.

Metric	Value
Samples	5342
Features (total)	16
Positives	751
Positive rate	0.1406
Imbalance (neg/pos)	6.11
Split policy	five repeated stratified i.i.d. train/test splits
Test size	0.2
CV folds	5

Table I reports dataset statistics and the evaluation protocol, while Table II lists the dataset features used in this study.

B. Baselines, Metrics, and Timing

The proposed Forward–Forward model is compared against logistic regression, random forests, and multi-layer perceptrons trained with back-propagation. Two MLP configurations are included, a smaller network and a depth-matched variant aligned with the Forward–Forward architecture, enabling a controlled comparison between learning rules under comparable capacity. Inputs are standardized with parameters fit on the training data for each split.

Performance is reported using PR-AUC, ROC-AUC, and F1-score at a fixed threshold (0.5). Wall-clock training time and inference time per sample are reported. Inference time per sample is computed as the wall-clock time to score the full test set divided by the number of test instances, with GPU synchronization when applicable.

Timing measurements are obtained on a single GPU machine with an AMD EPYC 9754 CPU, 755 GiB RAM, and an NVIDIA RTX 6000 Ada Generation GPU. Because neural and non-neural models were timed on different compute backends, these timing values should be interpreted as indicative throughput measurements rather than strict cross-backend speed comparisons.

Table III reports the main Forward–Forward hyperparameters used in the reported experiments. Table IV reports the main baseline hyperparameters.

V. RESULTS

A. Quantitative Comparison

Table V reports mean and standard deviation across five repeated stratified train/test splits. On this dataset, random forests achieve the strongest overall performance, with PR-AUC 0.9535 ± 0.0161 and F1 0.8683 ± 0.0169. Forward–

TABLE II
FEATURES IN THE ACEA DATASET, WITH GROUPING INTO CONSTITUTIVE PARAMETERS (CP) AND EXOGENOUS CAUSES (EC). FEATURE DESCRIPTIONS FOLLOW THE DATASET DOCUMENTATION REPORTED IN [13].

Feature	Type	Description
Minute (`CP_Minute-Fault`)	CP	Minute of the day when the fault was detected
Cable Section (`CP_Cable-section`)	CP	Section of the cable involved in the fault
Cable Length (`CP_Cable-length`)	CP	Length of the cable segment involved in the fault
Feeder Voltage (`CP_MV-voltage`)	CP	Nominal voltage of the feeder
Copper Percentage (`CP_Copper-percentage`)	CP	Percentage of copper in the cable
Air Percentage (`CP_Air-percentage`)	CP	Percentage of the feeder exposed to air
Level (`CP_Level`)	CP	Ordinal number of the secondary substation upstream of the faulty section
Level_Tot (`CP_Level_Tot`)	CP	Total number of secondary substations in the considered MV feeder
Semi-backbone MV Line Current (`EC_semi-b-MV-line-current`)	EC	Last current measured on the line riser before the fault event
Current (`EC_Current`)	EC	Estimate of the line current in the faulty branch based on normal operational data
Minimum Temperature (`EC_Min.-Temp.`)	EC	Minimum temperature recorded at the fault location
Maximum Temperature (`EC_Max.-Temp.`)	EC	Maximum temperature recorded at the fault location
Rain (`EC_Mm-rain`)	EC	Rainfall measured at the nearest weather station
Max Current (`EC_Max.-Current`)	EC	Maximum current recorded daily on the riser
Min Current (`EC_Min.-Current`)	EC	Minimum current recorded daily on the riser
Mean Current (`EC_Mean.-Current`)	EC	Average current recorded daily on the riser

TABLE III
FORWARD–FORWARD EXPERIMENTAL SETTINGS. SINGLE-FACTOR ABLATIONS REMOVE EITHER LAYERNORM OR PER-LAYER INPUT ℓ_2 NORMALIZATION, WITH ALL OTHER SETTINGS UNCHANGED.

Setting	Value
Architecture	label embedding, hidden sizes $(256, 256, 128)$, ReLU activations
Optimizer	layer-local Adam, learning rate 10^{-3}, weight decay 0
Epochs, batch size	30 epochs, batch size 256
Goodness and objective	$g_\ell = \|\mathbf{h}_\ell\|_2^2/d_\ell$, symmetric softplus loss, threshold $T = 1.0$
Normalization	LayerNorm per layer
Per-layer input normalization	enabled
Negative samples	deterministic binary label-flip

TABLE IV
BASELINE EXPERIMENTAL SETTINGS.

Model	Main settings
Logistic regression	`lbfgs` solver, $C = 1.0$, class weight balanced, max iter 2000
Random forest	500 trees, max depth unrestricted, class weight `balanced_subsample`
BP-MLP small	hidden sizes $(128, 64)$, AdamW, learning rate 10^{-3}, weight decay 10^{-4}, 50 epochs, batch size 256, early stopping with patience 10
BP-MLP matched	hidden sizes $(256, 256, 128)$, AdamW, learning rate 10^{-3}, weight decay 10^{-4}, 50 epochs, batch size 256, early stopping with patience 10

Forward remains competitive, reaching PR-AUC 0.9082 ± 0.0174 and F1 0.8476 ± 0.0146.

Forward–Forward improves over logistic regression in PR-AUC and F1-score and achieves higher F1-score than the depth-matched back-propagation MLP, while remaining close in ROC-AUC and PR-AUC. These results support the compatibility of goodness-based, layer-local training with imbalanced tabular fault detection, even when performance is benchmarked against strong, widely used baselines.

From a predictive maintenance perspective, PR-AUC is particularly informative because fault cases are rare and false alarms have operational costs. The depth-matched back-propagation baseline provides a controlled comparison in which differences are largely attributable to the learning rule rather than capacity.

Timing in Table V highlights an additional trade-off that matters in operational settings. Tree ensembles provide strong accuracy but exhibit substantially higher per-sample inference time under the present CPU timing setup, whereas neural models yield sub-millisecond inference under single-GPU execution with batched scoring. Forward–Forward remains within the same practical computational regime as the depth-matched back-propagation MLP, supporting its practical viability when both predictive quality and throughput are considered.

B. Precision–Recall Curves

Figure 2 reports Precision–Recall curves aggregated over seeds. Forward–Forward is competitive with the back-propagation baselines across a wide range of recall values, which is relevant for tuning the operating point based on the relative cost of missed faults versus unnecessary interventions. Ablation variants highlight that the method is sensitive to stabilization choices. Removing LayerNorm substantially degrades the Precision–Recall curve, while removing per-layer input normalization yields curves close to the main model. These effects are quantified in Section V-D.

C. Goodness Dynamics

Forward–Forward training optimizes a layer-local objective that separates positive from negative samples in terms of goodness. Figure 3 reports the evolution of the goodness margin, averaged across layers, batches, and seeds, defined as $\overline{g^+ - g^-}$. A positive margin indicates that activations induced by positive samples achieve higher goodness than their negative counterparts, which is the intended training signal. The same plot includes ablation variants to highlight the role of stabilization components.

Across seeds, the Trifecta-inspired model rapidly increases the margin and reaches a value of approximately 2.88 by the final epoch, while the no-LayerNorm variant remains near 0.36. This failure to build a substantial margin is consistent

TABLE V
TEST PERFORMANCE (MEAN±STD) ACROSS FIVE REPEATED STRATIFIED TRAIN/TEST SPLITS. MODELS MARKED WITH † ARE TRAINED AND EVALUATED ON A SINGLE GPU, WHILE THE REMAINING BASELINES ARE RUN ON CPU.

Model	PR-AUC	ROC-AUC	F1	Train(s)	Infer(ms/sample)
Forward–Forward (Trifecta)†	0.9082 ± 0.0174	0.9746 ± 0.0061	0.8476 ± 0.0146	2.86 ± 0.01	0.00108 ± 0.00015
Random Forest	0.9535 ± 0.0161	0.9910 ± 0.0026	0.8683 ± 0.0169	2.44 ± 0.01	0.22767 ± 0.00517
BP-MLP (small)†	0.9184 ± 0.0198	0.9801 ± 0.0058	0.8123 ± 0.0143	1.97 ± 0.50	0.00043 ± 0.00014
BP-MLP (matched)†	0.9180 ± 0.0145	0.9808 ± 0.0055	0.8038 ± 0.0140	0.99 ± 0.27	0.00038 ± 0.00013
Logistic Regression	0.8643 ± 0.0153	0.9630 ± 0.0029	0.7509 ± 0.0029	0.0074 ± 0.0007	0.00014 ± 0.00000

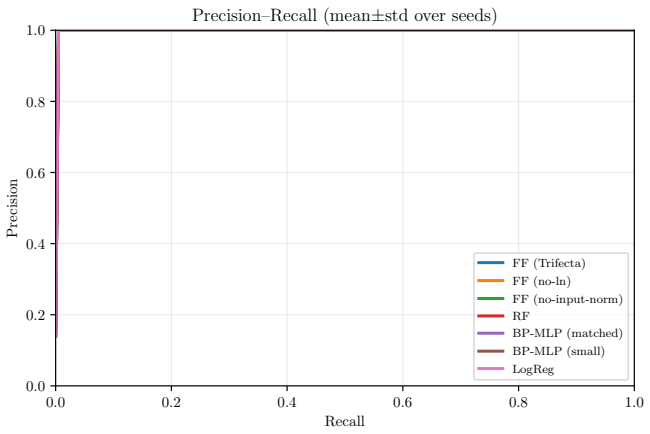

Fig. 2. Precision–Recall curves (mean±std) across five repeated stratified train/test splits.

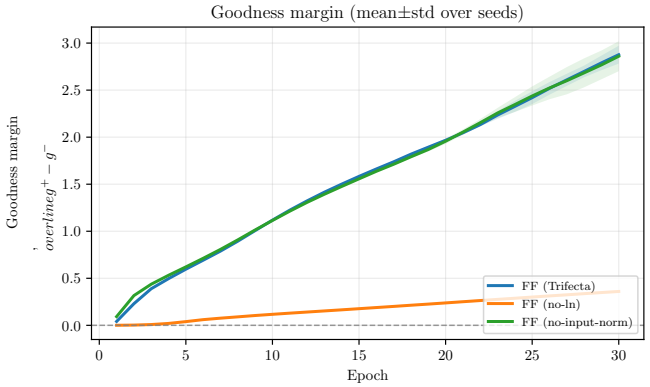

Fig. 3. Goodness margin $\overline{g^+ - g^-}$ over epochs, averaged across layers (mean±std across seeds).

with the downstream behavior observed in Table VI. The no-LayerNorm variant produces scores that do not cross the fixed decision threshold (0.5), which results in zero predicted positives and therefore $F1 = 0$.

D. Ablation Study

To isolate the effect of stabilization choices, an ablation is performed by disabling LayerNorm and per-layer input normalization in two separate variants, while keeping the same architecture, training protocol, and objective. For each seed, the delta is computed on the test split as the difference between the ablated variant and the main Forward–Forward model, and then aggregated across seeds as mean±std. Table VI reports these paired deltas.

TABLE VI
PAIRED MEAN±STD DELTAS OVER SEEDS, COMPUTED AS ABLATED VARIANT MINUS THE MAIN FF MODEL ON THE TEST SPLIT.

Variant	ΔPR-AUC	ΔROC-AUC	ΔF1	ΔTrain(s)
FF (no LN)	-0.544±0.043	-0.145±0.022	-0.848±0.015	-0.27±0.09
FF (no input norm)	0.010±0.006	0.003±0.003	0.005±0.015	0.39±0.79

TABLE VII
CALIBRATION METRICS (MEAN±STD) OVER SEEDS. LOWER VALUES INDICATE BETTER CALIBRATION.

Model	ECE	Brier
FF (Trifecta)	0.03757 ± 0.00433	0.03811 ± 0.00427
Random Forest	0.04125 ± 0.00374	0.02862 ± 0.00209
BP-MLP (matched)	0.06809 ± 0.00527	0.04905 ± 0.00357
Logistic Regression	0.11707 ± 0.00522	0.06808 ± 0.00430

Disabling LayerNorm causes a large drop in PR-AUC and yields $F1 = 0$ at the fixed threshold (0.5), which indicates that the model collapses to predicting no positives under strong imbalance. Disabling per-layer input normalization has a minor effect on PR-AUC and F1 while increasing training time. Overall, LayerNorm appears critical for stable Forward–Forward training on this dataset, while per-layer input normalization is less decisive under the present setting.

E. Probability Calibration

In predictive maintenance, probability-like fault scores enable risk-aware decisions, and their calibration is therefore relevant in addition to discriminative performance [14]. Table VII reports calibration metrics aggregated over seeds, including the Expected Calibration Error (ECE) and the Brier score. Figure 4 reports a reliability diagram for representative models.

Random forests achieve the lowest Brier score, while Forward–Forward attains lower ECE than the back-propagation MLP and logistic regression, indicating comparatively better calibration among neural baselines.

F. Discussion

Results indicate that Forward–Forward achieves competitive discriminative performance under class imbalance and stable

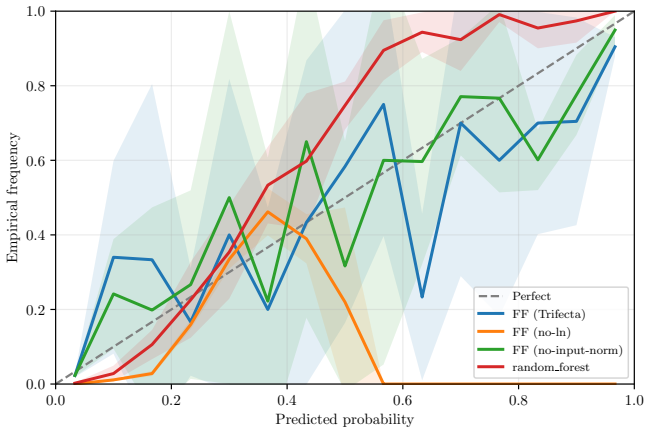

Fig. 4. Reliability diagram aggregated over seeds.

behavior across varying stratified splits. Random forests remain strong tabular baselines for structured data, while the depth-matched back-propagation MLP supports a controlled comparison in which differences are primarily attributable to the learning rule rather than model capacity. Goodness-margin dynamics provide a mechanistic view of training and help interpret downstream metrics in terms of the intended separation between positive and negative samples.

The ablation study performed suggests that stabilization is not a minor implementation detail on this dataset. LayerNorm appears critical for robust training, whereas per-layer input normalization has a smaller effect under the present setting. Several limitations should be noted. The study focuses on a single real-world dataset and a binary fault formulation, and it adopts an i.i.d. evaluation protocol rather than time-aware splits. The probabilistic outputs are obtained directly from goodness-based softmax without post-hoc calibration such as temperature scaling, and decision metrics are reported at a fixed threshold rather than optimized for operational cost. These constraints do not affect the main conclusion that stabilized Forward–Forward is viable for imbalanced tabular predictive maintenance, but they identify clear directions for strengthening deployment-oriented evaluations.

VI. Conclusion

This study evaluates a stabilized Forward–Forward learning formulation for imbalanced tabular predictive maintenance on a real-world fault dataset. Competitive performance is observed relative to classical baselines and back-propagation-trained neural networks under a standard stratified evaluation protocol with split variability across seeds. Future developments also aligned with predictive maintenance include evaluating time-aware splitting protocols and external validation on additional grid segments or operators, extending the formulation to multi-class and hierarchical fault taxonomies, and integrating cost-sensitive decision rules that map calibrated fault probabilities to inspection policies. On the algorithmic side, it is of interest to study Trifecta components not used here such as overlapping updates, to analyze robustness under distribution shift and concept drift, and to explore lightweight post-hoc calibration methods such as temperature scaling for improved probability reliability in deployment settings.

Acknowledgments

This study was carried out within the NEST - Network 4 Energy Sustainable Transition - and received funding from the European Union – NextGenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[2] G. Hinton, "The forward-forward algorithm: Some preliminary investigations," 2022. [Online]. Available: https://arxiv.org/abs/2212.13345

[3] F. Crick and G. Mitchison, "The function of dream sleep," *Nature*, vol. 304, no. 5922, pp. 111–114, 1983.

[4] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016. [Online]. Available: https://arxiv.org/abs/1607.06450

[5] T. Dooms, I. J. Tsang, and J. Oramas, "The trifecta: Three simple techniques for training deeper forward-forward networks," 2023. [Online]. Available: https://arxiv.org/abs/2311.18130

[6] A. Ororbia and A. Mali, "The predictive forward-forward algorithm," 2023. [Online]. Available: https://arxiv.org/abs/2301.01452

[7] R. Scodellaro, A. Kulkarni, F. Alves, and M. Schröter, "Training convolutional neural networks with the forward–forward algorithm," *Scientific Reports*, vol. 15, p. 38461, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-26235-2

[8] P. Gross, A. Boulanger, M. Arias, D. Waltz, P. M. Long, C. Lawson, R. Anderson, M. Koenig, M. Mastrocinque, W. Fairechio, J. A. Johnson, S. Lee, F. Doherty, and A. Kressner, "Predicting electricity distribution feeder failures using machine learning susceptibility analysis," in *Proceedings of the Eighteenth Innovative Applications of Artificial Intelligence Conference (IAAI)*, 2006.

[9] C. Rudin, D. Waltz, R. N. Anderson, A. Boulanger, A. Salleb-Aouissi, M. Chow, H. Dutta, P. N. Gross, B. Huang, S. Ierome, D. F. Isaac, A. Kressner, R. J. Passonneau, A. Radeva, and L. Wu, "Machine learning for the new york city power grid," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 328–345, 2012.

[10] M. Aghahadi, A. Bosisio, A. Pegoiani, S. Forciniti, M. Merlo, and A. Berizzi, "Predicting faults in power distribution grids during heatwaves: A comparative study of machine learning models applied to milan distribution network," *Sustainable Energy, Grids and Networks*, vol. 43, p. 101741, 2025.

[11] D. Kumar, U. Goswami, H. Kodamana, M. Ramteke, and P. K. Tamboli, "Variance-capturing forward-forward autoencoder (vffae): A forward learning neural network for fault detection and isolation of process data," *Process Safety and Environmental Protection*, vol. 178, pp. 176–194, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957582023006857

[12] D. Kumar, M. Ramteke, and H. Kodamana, "A framework for model maintenance using kernel-based forward propagating neural networks," *Chemical Engineering Research and Design*, vol. 189, pp. 352–364, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263876224005318

[13] E. De Santis, G. Ferro, and A. Rizzi, "A kan-shap framework for fault detection and analysis in smart grids," in *International Joint Conference on Neural Networks (IJCNN)*, 2025.

[14] A. Martino, E. De Santis, L. Baldini, and A. Rizzi, "Calibration techniques for binary classification problems: A comparative analysis," in *Proceedings of the 11th International Joint Conference on Computational Intelligence*, 2019, pp. 487–495.