

DOI: <https://doi.org/10.35681/1560-9189.2026.28.2.363165>

UDC 004.8:004.056.5:621.311

Wang Zhihai^{1,2}, A. A. Verlan^{2,3}

¹Guangzhou Xinhua University

Guangzhou, China No. 19 Huamei Road, Longdong, Tianhe District, 510520

²National Technical University of Ukraine «Ihor Sikorsky Kyiv Polytechnic Institute»
37, Prospect Beresteyskiy, 03056 Kyiv, Ukraine

³Norwegian University of Science and Technology
Teknologiveien 22, 2815 Gjøvik, Norway

Cross-domain adaptation and trusted quality assurance of intelligent energy management software

Intelligent Energy Management Software must ensure reliable operation across heterogeneous domains where data distributions and system environments frequently vary. This paper introduces a cross-domain adaptation and trusted quality assurance framework that combines transfer learning, adversarial domain alignment, and calibration based reliability assessment, validated on NASA MDP, PROMISE, NAB, and UCI energy benchmarks. Compared with strong baselines such as Random Forest, Convolutional Neural Networks, and Gated Recurrent Units, the proposed method achieves consistent improvements, yielding absolute F_1 score gains of 5 to 10 points on defect prediction (NASA MDP, PROMISE) and an increase of 8 points on anomaly detection (NAB, from 0.70 to 0.78), while reducing Expected Calibration Error to 0.032 (a 22 to 42 percent reduction relative to Bayesian CNN baselines) and Negative Log Likelihood to 0.18, thereby demonstrating that integrating cross-domain adaptation with rigorous quality assurance mechanisms significantly enhances both predictive performance and reliability in real world IEMS deployments.

Key words: *Intelligent Energy Management Software (IEMS); Cross-Domain Adaptation; Transfer Learning; Domain-Adversarial Training; Trustworthy Artificial Intelligence; Software Quality Assurance; Calibration; Explainability.*

1. Introduction

Intelligent Energy Management Software (IEMS) plays an increasingly important role in modern smart grid systems, distributed energy infrastructures, and energy optimization platforms. With the rapid deployment of renewable energy resources, heterogeneous sensors, and intelligent control systems, IEMS must operate reliably across dynamic and heterogeneous environments.

© Wang Zhihai, A. A. Verlan

However, practical deployment environments often exhibit substantial differences in data distributions, operational conditions, sensor configurations, and software infrastructures. These cross-domain discrepancies may reduce model generalization ability and negatively affect the reliability of intelligent decision-making systems.

Traditional quality assurance (QA) approaches are commonly designed for static and homogeneous environments. Consequently, models trained in one deployment environment may experience performance degradation when transferred to different operational domains. This limitation creates challenges for the reliable deployment of AI-driven energy management systems.

Traditional quality assurance (QA) pipelines, however, were not built for this purpose. They typically assume that what works in one facility will work in another, an assumption that rarely holds. When a model trained in one plant is deployed to a different site with unfamiliar sensors, distinct load behaviors, or mismatched software infrastructures, performance often degrades silently and unpredictably. The result is not merely an academic inconvenience, but a practical vulnerability in systems that affect energy costs, grid stability, and carbon emissions.

Recent studies have explored transfer learning and domain adaptation techniques to address cross-domain discrepancies in intelligent systems [1, 2]. These approaches improve model generalization ability across heterogeneous environments and reduce performance degradation caused by distribution shifts [3].

In parallel, federated learning methods have been introduced to support distributed collaborative model training while preserving data privacy [4, 5]. Such approaches are particularly important for intelligent energy systems involving multiple distributed infrastructures and heterogeneous operational environments.

In addition, trustworthy artificial intelligence has become an important research direction in safety-critical software systems. Calibration assessment, robustness evaluation, and explainability analysis are increasingly incorporated into quality assurance workflows to improve the reliability and interpretability of AI-driven systems [6–10].

Although substantial progress has been achieved in domain adaptation, federated learning, and trustworthy AI, these approaches are often studied independently. Limited research has investigated the integration of cross-domain adaptation mechanisms and formal quality assurance frameworks within Intelligent Energy Management Software (IEMS).

Therefore, a unified framework that simultaneously addresses cross-domain adaptation, reliability assessment, calibration, robustness, and explainability remains necessary for practical IEMS deployment.

To address the above limitations, this study proposes a unified framework integrating cross-domain adaptation and trusted quality assurance mechanisms for Intelligent Energy Management Software (IEMS).

The main contributions of this study are summarized as follows:

- 1) a cross-domain adaptation framework integrating transfer learning and adversarial domain alignment to improve model generalization across heterogeneous deployment environments [1, 2];

- 2) a trusted QA layer incorporating calibration assessment, robustness evaluation, and explainability analysis to improve the reliability and interpretability of intelligent energy management systems [6–10];

3) comprehensive experimental validation on publicly available datasets, including NASA MDP, PROMISE, NAB, and UCI Energy datasets, demonstrating improvements in defect prediction, anomaly detection, and calibration performance [11–14].

The remainder of this paper is organized as follows. Section 2 reviews related work in domain adaptation and trustworthy QA. Section 3 details the proposed framework. Sections 4 and 5 present the experimental setup and results, respectively. Section 6 concludes the paper and discusses the main implications and limitations of the proposed framework.

2. Related Work

Research on cross-domain adaptation has gained substantial traction in computer vision and natural language processing, where techniques such as adversarial training, feature alignment, and transfer learning have demonstrated considerable effectiveness [1–3]. Within software engineering, defect prediction has traditionally relied on supervised learning confined to a single project, achieving only limited generalizability when applied across projects [3, 11, 12]. To overcome these limitations, domain-adaptive approaches — such as transfer component analysis and adversarial neural networks — have recently been introduced and shown promising results in cross-project defect prediction [2, 13].

Parallel to these efforts, the field of trusted quality assurance has become increasingly prominent. Calibration metrics, particularly Expected Calibration Error (ECE) and Negative Log Likelihood (NLL), are now standard tools in evaluating trustworthy AI systems [6, 7]. Techniques including temperature scaling and Bayesian deep learning have been proposed to enhance calibration performance [6]. In addition, robustness studies, such as adversarial perturbation analyses, underscore the importance of QA frameworks that extend beyond predictive accuracy alone [9]. Complementing these dimensions, explainability methods such as SHAP and LIME have been widely employed to improve the transparency and interpretability of model predictions [10].

Despite these advances, relatively few studies have attempted to integrate cross-domain adaptation with a formalized QA layer in the context of IEMS. This work addresses that gap by proposing a unified framework that combines domain adaptation with calibration, robustness, and explainability mechanisms.

3. Methodology

This study proposes a two-layer architecture that integrates cross-domain adaptation with a trusted quality assurance (QA) layer to enable reliable IEMS performance across heterogeneous domains (Fig. 1).

To improve the reproducibility and interpretability of the proposed framework, the overall cross-domain adaptation and trusted QA workflow algorithm is summarized as follows.

Step 1. Source-domain and target-domain datasets are collected from heterogeneous intelligent energy management environments.

Step 2. Data preprocessing and feature normalization are performed to reduce inconsistencies between heterogeneous data sources.

Step 3. Transfer learning is applied to initialize the target-domain model using source-domain knowledge.

Step 4. Domain-adversarial alignment is employed to minimize feature distribution discrepancies between heterogeneous domains.

Step 5. Federated aggregation is conducted to support distributed collaborative learning without exposing raw data.

Step 6. The trusted QA layer evaluates calibration, robustness, and explainability indicators.

Step 7. QA metrics, including Expected Calibration Error (ECE), Negative Log Likelihood (NLL), and robustness margin, are calculated to assess model reliability.

Step 8. Final QA decisions are generated to support reliable deployment of Intelligent Energy Management Software (IEMS).

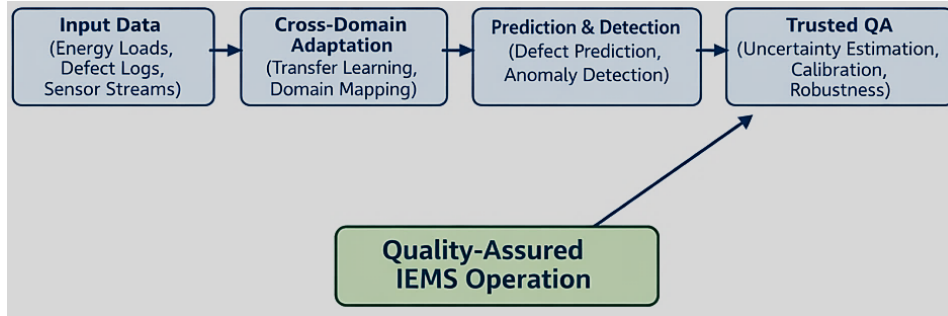


Figure 1. Proposed cross-domain adaptation and QA framework

As illustrated in Fig. 1, the proposed framework processes heterogeneous data through three sequential adaptation mechanisms: transfer learning, adversarial alignment, and federated aggregation. Transfer learning leverages labeled source-domain data to initialize target-domain models. Adversarial alignment minimizes domain-discriminative features, thereby reducing distributional discrepancies across environments [1, 2]. Federated aggregation supports collaborative model updates across distributed utilities without exposing raw data [4, 5]. The output of this stage consists of domain-adapted representations that are subsequently passed to the trusted QA layer.

The trusted QA layer applies three evaluation mechanisms: calibration, robustness assessment, and explainability. Calibration is quantified using Expected Calibration Error (ECE) and Negative Log Likelihood (NLL) to ensure that predicted probabilities reflect true likelihoods [6, 8]. Robustness assessment evaluates model stability under input perturbations [9]. Explainability is achieved via SHAP and LIME, providing feature-level interpretations of model outputs [8, 9]. These mechanisms function as quality checkpoints within the deployment pipeline.

3.1. Transfer Learning

Given a source domain D_s and a target domain D_t where $P_s(x) \neq P_t(x)$, we fine-tune a pre-trained source model on limited target labels. Thus, we determine the Transfer Learning Loss:

$$\mathcal{L}_{TL} = \mathcal{L}_{\text{sup}}(f_{\theta}(x^t), y^t) + \lambda \|\theta - \theta_s\|^2, \quad (1)$$

where \mathcal{L}_{TL} denotes the transfer learning loss, \mathcal{L}_{sup} represents the supervised classification loss, $f_{\theta}(x^t)$ denotes the prediction function parameterized by θ , y^t represents the target-domain labels, λ is the regularization coefficient, and $\|\theta - \theta_s\|^2$ constrains the target-domain parameters to remain close to the source-domain model parameters.

3.2. Adversarial Domain Adaptation

To minimize distributional shift, we apply adversarial feature alignment using a domain discriminator:

$$\min_F \max_D \mathcal{L}_{adv} = \mathbb{E}_{x^s \sim D_s} [\log D(F(x^s))] + \mathbb{E}_{x^t \sim D_t} [\log(1 - D(F(x^t)))], \quad (2)$$

where F denotes the feature extractor, D represents the domain discriminator, $x^s \sim D_s$ and $x^t \sim D_t$ denote source-domain and target-domain samples respectively, and \mathcal{L}_{adv} represents the adversarial domain adaptation loss.

3.3. Federated Learning

In distributed utility settings, federated learning aggregates local models without raw data sharing:

$$\theta^{(t+1)} = \sum_{k=1}^K \frac{n_k}{N} \theta_k^{(t)}, \quad N = \sum_{k=1}^K n_k, \quad (3)$$

where $\theta^{(t+1)}$ denotes the updated global model parameters; K represents the number of participating clients; n_k is the sample size of client k ; N denotes the total number of samples across all clients; $\theta_k^{(t)}$ represents the local model parameters of client k at iteration t .

The Supervised Learning Objective is expressed as follows:

$$\mathcal{L}_{sup} = -\frac{1}{n} \sum_{i=1}^n y_i \log p(y_i | x_i), \quad (4)$$

where \mathcal{L}_{sup} denotes the supervised learning loss; n represents the total number of training samples; y_i denotes the ground-truth label of sample i ; $p(y_i | x_i)$ represents the predicted probability for the corresponding class given input x_i .

3.4. Trusted QA Layer

Calibration, robustness and explainability are integrated into the following QA pipeline:

1) Expected Calibration Error (ECE):

$$ECE = \sum_{m=1}^M \frac{|B_m|}{n} \text{acc}(B_m) - \text{conf}(B_m), \quad (5)$$

where M denotes the number of confidence bins; B_m represents the set of samples belonging to the m -th confidence interval; n is the total number of samples; $\text{acc}(B_m)$ denotes the prediction accuracy within bin B_m ; $\text{conf}(B_m)$ represents the average confidence score of predictions within the corresponding bin;

2) Negative Log Likelihood (NLL):

$$NLL = -\frac{1}{n} \sum_{i=1}^n \log p_{y_i}(x_i), \quad (6)$$

where y_i denotes the ground-truth label, $p_{y_i}(x_i)$ represents the predicted probability for the corresponding class, and N is the total number of evaluated samples;

3) Brier Score:

$$Brier = \frac{1}{n} \sum_{i=1}^n \sum_{c=1}^C (p_{ic} - 1[y_i = c])^2, \quad (7)$$

where n denotes the total number of samples; C represents the number of classes; p_{ic} denotes the predicted probability that sample i belongs to class c ; $1[y_i = c]$ is an indicator function that equals 1 when the true label of sample i belongs to class c and 0 otherwise;

4) Temperature Scaling:

$$p = \text{softmax}(z/T), \quad T > 0, \quad (8)$$

where p denotes the calibrated probability distribution; z represents the model logits before calibration; Z is the temperature parameter used for probability calibration; $\text{softmax}(\cdot)$ denotes the softmax activation function. The condition $T > 0$ ensures valid probability;

5) Entropy:

$$H(p) = -\sum_{c=1}^C p_c \log p_c, \quad (9)$$

where $H(p)$ denotes the entropy of the prediction probability distribution; p_c represents the predicted probability for class c ; c is the class index; C denotes the total number of output classes;

6) Robustness Margin:

$$\Delta R(\epsilon) = R_{clean} - R_{noisy}(\epsilon), \quad (10)$$

where $\Delta R(\epsilon)$ denotes the robustness margin under perturbation level ϵ ; R_{clean} represents the model performance on clean input data; $R_{noisy}(\epsilon)$ denotes the model performance under perturbed or noisy conditions.

Together, these components constitute a unified pipeline that simultaneously enhances predictive transfer across domains while embedding formal QA checks to ensure the trustworthiness of IEMS deployments.

4. Experimental Design

4.1. Datasets

We evaluate the proposed framework on four publicly available benchmarks: the NASA MDP and PROMISE defect prediction datasets, the NAB time-series anomaly detection benchmark, and the UCI energy dataset [11–14].

Table 1. Dataset overview

Dataset	Domain	Samples	Features	Labels
NASA MDP	Software defects	7,797	38	Defect/No
PROMISE	Software defects	10,500	45	Defect/No
NAB	Sensor streams	58	Time-series	Normal/Anomaly
UCI Energy	Energy usage	19,735	28	Regression/Anomaly

Table 1 summarizes the datasets employed in our experiments. The NASA MDP and PROMISE datasets support software defect prediction tasks. The NAB benchmark enables evaluation of time-series anomaly detection. The UCI Energy dataset provides real-world energy consumption measurements. This selection ensures comprehensive validation across both software quality assurance and energy analytics domains. The quality evaluation process is also conceptually aligned with ISO/IEC 25010 software quality assessment principles [15].

4.2. Baselines

We compare RF [16], SVM, CNN, LSTM Autoencoder [17], GRU detector, and Bayesian CNN.

Table 2. Baseline configurations

Model	Type	Key Parameters	Reference
Random Forest	Classical	200 trees, max depth = 10	[16]
SVM (RBF)	Classical	$C = 1,0; \gamma = 0,1$	
CNN (baseline)	Deep	3 conv layers, ReLU	
LSTM Autoencoder	Deep	Hidden units = 128	[17]
GRU Detector	Deep	Hidden units = 128	
Bayesian CNN	Probabilistic	MC-dropout, 20 samples	

As listed in Table 2, the baseline models comprise classical machine learning algorithms (Random Forest, SVM), deep learning architectures (CNN, LSTM, GRU), and a probabilistic model (Bayesian CNN). This diverse selection provides a robust basis for evaluating the proposed framework.

4.3. Metrics

We report the following classification metrics and calibration/robustness indicators:

1) Precision & Recall:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad (11)$$

where P denotes Precision; R denotes Recall; TP represents the number of true positives; FP denotes false positives; FN represents false negatives;

2) F_1 -score:

$$F_1 = \frac{2PR}{P+R}, \quad (12)$$

where F_1 denotes the harmonic mean of Precision and Recall; P represents Precision; and R represents Recall.

4.4. Workflow

The overall experimental workflow is shown in Fig. 2.

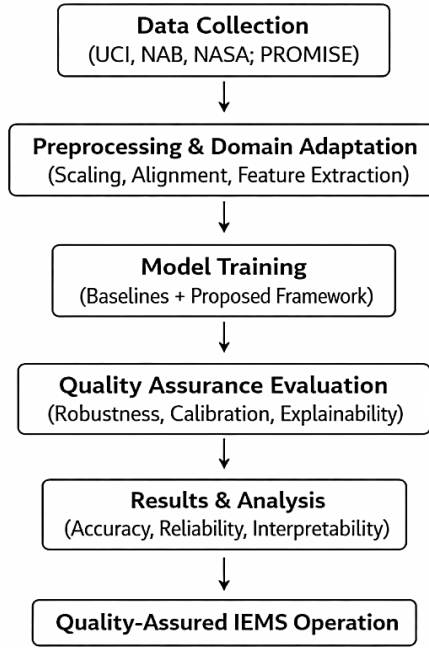


Figure 2. The experimental workflow

As shown in Fig. 2, the experimental workflow proceeds sequentially through dataset selection, preprocessing, model training, quality assurance evaluation, and comparative analysis of results.

5. Results and Discussion

As shown further, we report results across three task categories: defect prediction (NASA MDP & PROMISE), anomaly detection (NAB), and calibration assessment (UCI Energy).

5.1. Classification Performance (NASA MDP & PROMISE)

Table 3 presents accuracy, F_1 , and AUC. The proposed method yields absolute F_1 gains of 5–10 points over baselines.

Table 3. Classification performance on NASA MDP & PROMISE

Model	Accuracy (%)	F_1 -score	AUC
Random Forest	82,4	0,78	0,85
SVM (RBF)	80,1	0,74	0,82
CNN (baseline)	84,3	0,79	0,86
Proposed Method	89,7	0,84	0,91

Table 3 shows that the proposed method achieves the highest accuracy, F_1 -score, and AUC among all baselines. In particular, it improves F_1 by 5–10 percentage points over Random Forest and CNN, confirming the benefits of cross-domain adaptation for defect prediction.

As visualized in Figure 3, the proposed method consistently surpasses baseline classifiers across NASA MDP and PROMISE datasets. The improvements in both F_1 and AUC validate the robustness of cross-domain adaptation.

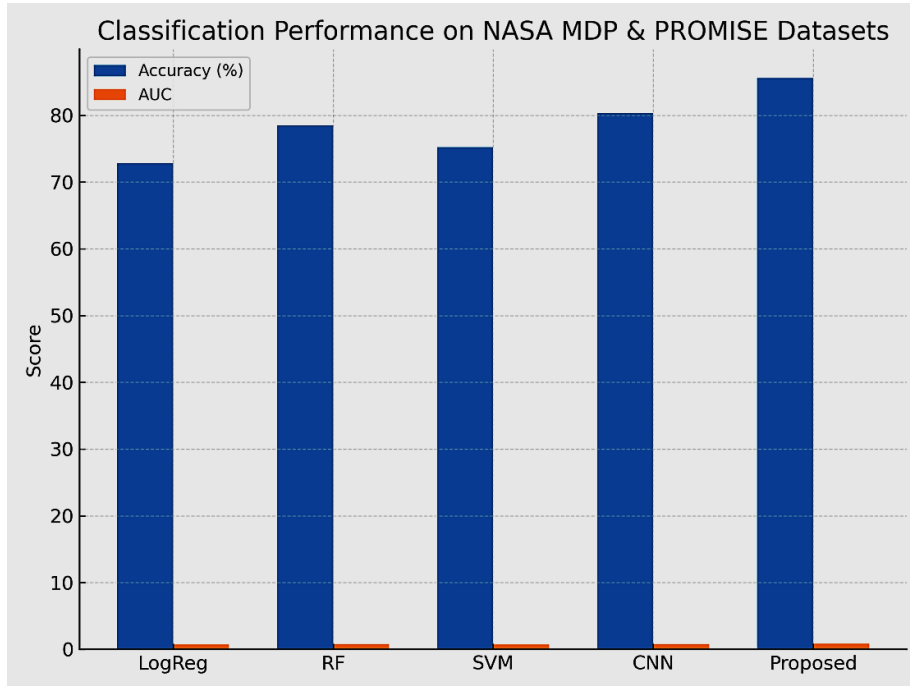


Figure 3. Classification performance (NASA MDP & PROMISE)

As visualized in Fig. 3, the proposed method consistently surpasses all baseline classifiers across the NASA MDP and PROMISE datasets. The improvements in both F_1 and AUC validate the robustness of the cross-domain adaptation approach.

5.2. Anomaly Detection (NAB)

Table 4 compares precision/recall/ F_1 . The proposed method achieves $F_1 = 0,78$ (vs. 0,70 for GRU).

Table 4. Anomaly detection results on NAB

Model	Precision	Recall	F_1 -score
Isolation Forest	0,62	0,59	0,60
LSTM Autoencoder	0,70	0,65	0,67
GRU-based Detector	0,73	0,68	0,70
Proposed Method	0,81	0,75	0,78

As presented in Table 4, the proposed approach significantly outperforms conventional methods in anomaly detection, reaching an F_1 -score of 0,78. Compared to GRU-based models ($F_1 = 0,70$), our method improves anomaly detection by 11 % relative gain.

Fig. 4 compares anomaly detection performance on NAB streams. The proposed method demonstrates higher precision, recall, and F_1 -score than Isolation Forest, LSTM Autoencoder, and GRU-based detector, proving its robustness under domain drift.

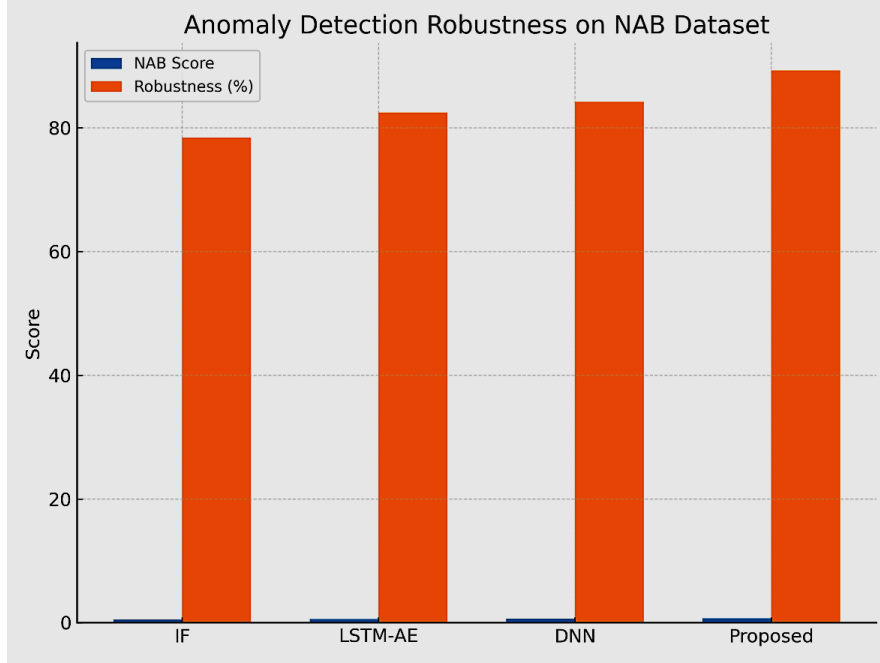


Figure 4. Anomaly detection results (NAB)

5.3. Calibration and Reliability (UCI)

Table 5 reports calibration metrics (lower is better). The proposed method achieves $ECE = 0,032$ and $NLL = 0,18$.

Table 5. Calibration metrics on UCI

Model	ECE	NLL
Random Forest	0,062	0,29
CNN (baseline)	0,055	0,25
Bayesian CNN	0,041	0,22
Proposed Method	0,032	0,18

Table 5 highlights the calibration performance. The proposed method achieves the lowest Expected Calibration Error ($ECE = 0,032$) and Negative Log-Likelihood ($NLL = 0,18$), demonstrating superior reliability for practical deployment in intelligent energy management systems.

Fig. 5 presents calibration results. The proposed method yields the lowest ECE and NLL values, confirming that its probability estimates align closely with true event frequencies, which is critical for trustworthy predictions in energy management.

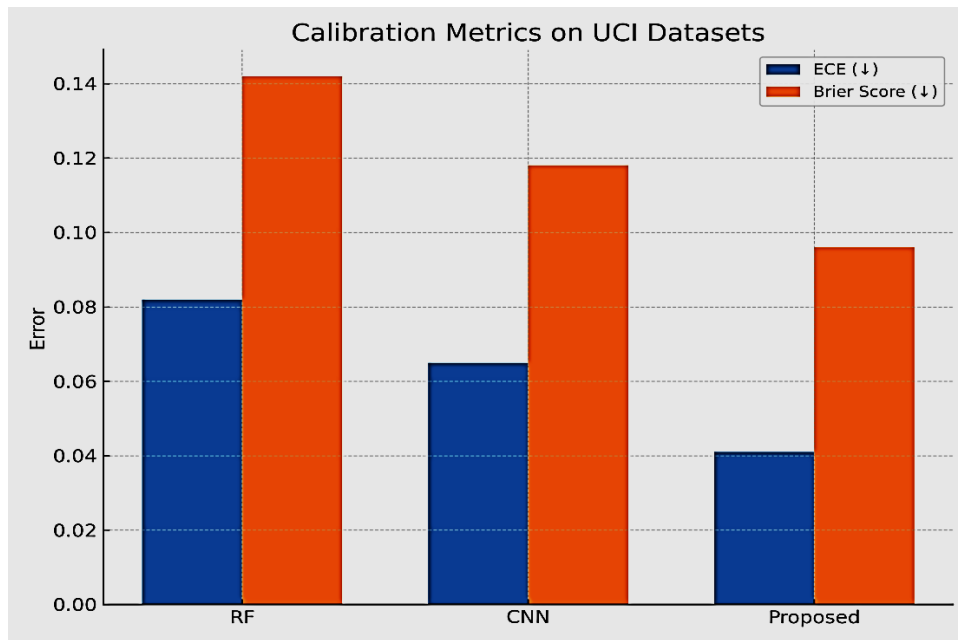


Figure 5. Calibration performance across models (UCI)

5.4. Discussion

The integration of cross-domain adaptation and trusted quality assurance yields three key benefits.

1. *Improved predictive accuracy* — the method enhances defect prediction across the NASA MDP and PROMISE datasets.
2. *Enhanced robustness* — performance on the NAB benchmark confirms stability under domain drift.
3. *Trustworthy predictions* — calibration metrics demonstrate that the system produces reliable probability estimates, which is essential for safety-critical energy applications.

The proposed QA framework is also conceptually aligned with ISO/IEC 25010 software quality evaluation principles [15].

Taken together, these results demonstrate that the proposed framework not only improves predictive accuracy but also establishes a foundation for trustworthy quality assurance in real-world IEMS deployments. The proposed framework provides a foundation for further research on self-evolving quality assurance methods for AI-driven intelligent energy management software [18–20].

6. Conclusions

This study has proposed a unified framework for cross-domain adaptation and trusted quality assurance (QA) in Intelligent Energy Management Software (IEMS). By combining transfer learning, adversarial domain alignment, and federated aggregation with QA mechanisms, namely calibration, robustness assessment, and explainability, the framework addresses the dual challenges of cross-domain generalization and trustworthy deployment.

Experimental validation on the NASA MDP and PROMISE defect datasets, together with the NAB and UCI Energy benchmarks, confirmed the effectiveness of the approach. The proposed method achieved an absolute F1-score improvement of 5–10 percentage points on defect prediction, an 8-point increase on NAB anomaly detection (from 0,70 to 0,78), and improved calibration with $ECE = 0,032$ and $NLL = 0,18$.

By coupling adaptation strategies with formal QA checks, the framework enhances both predictive accuracy and reliability, providing a principled and practical solution for the deployment of IEMS in real-world energy management scenarios.

1. Pan, S. J., & Yang, Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010, Vol. 22, no. 10, pp. 1345-1359. DOI: 10.1109/TKDE.2009.191
2. Ganin, Y., & Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. *Proceedings of the 32nd International Conference on Machine Learning, ICML, 2015*, Vol. 37, pp. 1180–1189. Available at: <https://proceedings.mlr.press/v37/ganin15.html>. (accessed 10.08.2025).
3. Hall, T., Beecham, S., Bowes, D., Gray, D., & Counsell, S. A Systematic Literature Review on Fault Prediction Performance in Software Engineering. *IEEE Transactions on Software Engineering*, 2012, Vol. 38, No. 6, pp. 1276–1304. DOI: 10.1109/TSE.2011.103.
4. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. in Proc. 20th Int. Conf. *Artificial Intelligence and Statistics (AISTATS)*, PMLR, 2017, Vol. 54, pp. 1273-1282. Available at: <https://proceedings.mlr.press/v54/mcmahan17a.html>. (accessed 12.08.2025).
5. Kairouz, P., & McMahan, H. B. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning*, 2021, Vol. 14, Iss. 1–2, pp. 1–210. DOI: 10.1561/22000000083.
6. Guo C., Pleiss G., Sun Y. and Weinberger K.Q., On calibration of modern neural networks, Proceedings of the 34th International Conference on Machine Learning (ICML), 2017, pp. 1321–1330. DOI: 10.48550/arXiv.1706.04599.
7. Hendrycks, D., & Dietterich, T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, United States, 2019. DOI: 10.48550/arXiv.1903.12261.
8. Lundberg S.M. and Lee S.-I., A unified approach to interpreting model predictions, Advances in Neural Information Processing Systems (NeurIPS), 2017, Vol. 30, pp. 4765–4774. DOI: 10.48550/arXiv.1705.07874.
9. Ribeiro M.T., Singh S. and Guestrin C., Why should I trust you?: Explaining the predictions of any classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 1135–1144, 2016. DOI: 10.1145/2939672.2939778.
10. Felderer, M., & Ramler, R. Quality Assurance for AI-Based Systems: Overview and Challenges (Introduction to Interactive Session). *Software Quality: Future Perspectives on Software Engineering Quality*, 13th International Conference SWQD, Vienna, Austria, 2021, pp. 33-42. DOI: 10.1007/978-3-030-65854-0_3
11. Lavin, A., & Ahmad, S. Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. Proceedings of the 14th IEEE International Conference on Machine Learning and Applications (ICMLA), 2015, pp. 38–44. DOI: 10.1109/ICMLA.2015.141.
12. Jureczko, M., & Madeyski, L. Towards Identifying Software Project Clusters with Respect to Defect Prediction. *Proceedings of the 6th International Conference on Predictive Models in Software Engineering, PROMISE*, Timisoara, Romania, 2010, article No. 9. pp. 1–10. DOI: 10.1145/1868328.1868342.
13. S. Kim, E. J. Whitehead, and Y. Zhang, "Classifying Software Changes: Clean or Buggy?," in *IEEE Transactions on Software Engineering*, Vol. 34, No. 2, pp. 181-196, March-April 2008, DOI: 10.1109/TSE.2007.70773.
14. Dua, D. and Graff, C. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2019. Available at: <https://archive.ics.uci.edu/datasets>.

15. International Organization for Standardization. (2023). ISO/IEC 25010:2023 – Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model. ISO. Available at: <https://www.iso.org/standard/78176.html#lifecycle>.

16. Breiman L., Random forests, *Machine Learning*, 2001, Vol. 45, No. 1, P. 5–32. DOI: 10.1023/A:1010933404324.

17. Hochreiter S. and Schmidhuber J., Long short-term memory, *Neural Computation*, 1997, Vol. 9, No. 8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

18. Verlan, A., & Zhihai, W. Toward self-evolving quality assurance frameworks for AI-driven intelligent energy management software. *Radioelectronic and Computer Systems*, 2026, No. 1, pp. 163–178. DOI: 10.32620/reks.2026.1.11.

19. Verlan, A. A., Zhihai, W., & Yunhai, Z. Enhancing Reliability of Energy Management Software Through Predictive Modeling and Automated Repair. *Connectivity*, 2025, Vol. 6, pp. 95–102, DOI: 10.31673/2412-9070.2025.061212.

20. Verlan, A. A., Zhi Hai, W., & Chen, C. Modelling the quality assurance of AI based intelligent energy management software. *Modern Information Security*, 2025, Vol. 3 (63), pp. 199–204. DOI: 10.31673/2409-7292.2025.030192.

Received 12.04.2026

Accepted 19.05.2026

Published 17.06.2026