 https://doi.org/10.1051/e3sconf/202672301015

Practical Real-Time AC Arc Fault Detection using Lightweight Features on Embedded Hardware

Tan Tong Sy[1,*], *Toan* Le Duc[1,**], *Dao* Le Xuan[2,***], *Anh* Do Duc[2,****], *Bien* Nguyen Quang[3,†], and *Duan* Luong Cong[3,‡]

[1]Faculty of Electronics Engineering 1,
Posts and Telecommunications Institute of Technology, Vietnam

[2]Rang Dong Light Source & Vacuum Flask JSC, Hanoi, Vietnam

[3]Faculty of Electronics Engineering 1 & EDA Lab,
Posts and Telecommunications Institute of Technology, Vietnam

[‡] **Correspondence:** e-mail: duanlc@ptit.edu.vn

Abstract. AC series arc faults in household appliances pose significant safety risks but remain challenging to detect due to load-dependent signatures and complex interactions in mixed-branch operations. This paper presents a practical, real-time arc fault detection system that integrates a comprehensive two-branch household dataset, lightweight handcrafted features, and classical machine learning for deployment on an STM32F407 microcontroller (MCU). Current signals were sampled at 10 kHz across 32 scenarios, encompassing 13 single-load and 19 two-branch mixed-load cases. Each signal window is represented by a 133-dimensional feature vector, comprising five statistical descriptors and 128 FFT magnitude coefficients, optimized for resource-constrained environments. Among various evaluated classifiers, Random Forest was selected for its superior trade-off between recognition performance and deployment efficiency. The proposed model achieved 89.18% accuracy, 91.34% recall, and 86.96% F1-score. Furthermore, a leave-one-scenario-out (LOSO) evaluation was employed to systematically analyze the masking effect in multi-branch environments, identifying challenges in which high-power loads obscure weak arc signatures. The complete firmware requires 896 KB of Flash and 102 KB of RAM, with a total processing time of only 7.02 ms per cycle, significantly within the update interval and confirming its feasibility for low-cost, real-time protection devices.

1 Introduction

Series AC arc faults are a major safety concern in low-voltage residential electrical systems because they can generate high temperature discharge while the circuit current may still remain below the trip threshold of conventional protection devices [1–3]. Detection becomes more difficult when the connected appliance is nonlinear or when several branches operate together, because normal load harmonics and switching behavior can resemble arc-related disturbances [4–7].

Many earlier studies used manual current features and rule-based or machine-learning classifiers to identify arc conditions [1, 4, 8]. More recent work has increasingly adopted convolutional neural networks, lightweight deep models, and transformer-based architectures to improve recognition accuracy [9–11]. These studies show strong progress, yet practical deployment still faces two persistent challenges. First, detection performance often drops when the load type changes or when mixed-load masking occurs. Second, high offline accuracy alone is not enough for a real protection device, because the full signal-processing and inference chain must run reliably on limited embedded hardware [13, 14].

From a practical engineering perspective, a residential arc detector must satisfy requirements beyond mere classification accuracy. It should operate reliably with current-only sensing, a moderate sampling frequency, and limited memory, while ensuring a stable runtime on a low-cost processor [15, 16]. By prioritizing a resource-efficient design, the proposed solution directly reduces the overall manufacturing costs of protective devices, thereby enhancing their economic feasibility and facilitating large-scale deployment in low-voltage residential systems.

This work studies a practical compromise between detection accuracy and embedded feasibility. Instead of using a high-complexity raw-signal model, the proposed system uses a compact set of statistical and spectral current features, benchmarks several conventional classifiers, and

[*] e-mail: tants.b22dt276@stu.ptit.edu.vn
[**] e-mail: toanld@ptit.edu.vn
[***] e-mail: xuandao.vnatr@gmail.com
[****] e-mail: anhdd@rangdong.com.vn
[†] e-mail: biennq@ptit.edu.vn
[‡] e-mail: duanlc@ptit.edu.vn

deploys the selected model on an STM32F407 MCU. The dataset is intentionally designed to include both single-load and two-branch mixed-load scenarios so that masking effects can be observed directly.

The main contributions of this paper are threefold. First, a household arc fault dataset with 32 scenarios is organized from current measurements, including 13 single-load arc cases and 19 two-branch mixed-load cases. Second, a lightweight 133-dimensional feature set is designed for practical embedded execution. Third, the full pipeline, from data acquisition to alarm indication, is implemented on MCU hardware and evaluated with both validation and LOSO testing.

The rest of this paper is organized as follows. Section 2 describes the dataset, feature extraction process, model development, embedded deployment, and evaluation metrics. Section 3 presents the benchmark, validation, LOSO, and prototype testing results. Section 4 discusses the main findings, their relation to prior work, and the current limitations of the proposed approach. Section 5 concludes the paper.

2 Methodology

2.1 Dataset and Preprocessing

Current signals were collected from household electrical appliances under arc and non-arc conditions. Each acquisition case lasted at least 3 min, and most cases lasted about 5 min. The sampling frequency was 10 kHz. The experimental setup and data collection device are shown in Figure 1.

The dataset contains 32 scenarios constructed from various electrical loads detailed in Table 1. Among them, 13 are single-load arc cases, and 19 are two-branch mixed-load cases in which the arc branch and the no-arc branch belong to different appliances. This two-branch design is important because it reflects practical branch interaction and makes masking effects visible during evaluation. After segmentation with a 4096-sample sliding window and a 2048-sample hop, the dataset produced 68,385 labeled windows, including 41,368 normal windows and 27,017 arc windows. With the 10 kHz sampling rate, the window length is about 409.6 ms and the update interval is about 204.8 ms.

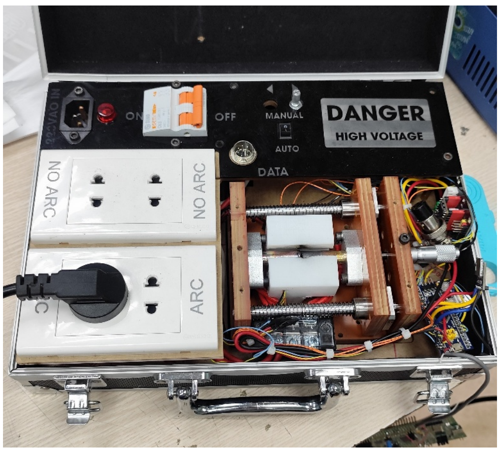

Figure 1: Experimental setup and data collection device for household arc fault measurements.

2.2 Feature Extraction

To keep the implementation suitable for a MCU, each signal window was transformed into a compact manual feature vector instead of using direct raw-waveform inference. The final representation contains 133 features, as detailed in Table 2. Five features describe the signal in the time domain, while 128 features describe the spectral content from the retained magnitude bins of a 256-point FFT. This design follows the practical direction of earlier multi-feature arc studies [5, 7, 17].

The selected features provide a good tradeoff between information content and computational cost. Mean, RMS, and standard deviation capture the energy level and dispersion of the current waveform. Skewness and kurtosis describe waveform shape changes caused by arc irregularity. The FFT magnitudes summarize the frequency-domain disturbance pattern, which is useful because arc events often introduce unstable spectral components [17–19].

2.3 Model Development and Training Strategy

Model development was carried out in three stages. First, several candidate classifiers were trained and compared to identify the most suitable model family. Second, the best candidate was tuned on a 70/30 train-validation split. Third, LOSO evaluation was performed by training on 31 scenarios and testing on the remaining one. This final stage was used to study generalization under unseen appliance conditions.

The compared models include Extreme Gradient Boosting, Random Forest, Extra Trees, Gradient Boosting, Linear Discriminant Analysis, Ridge Classifier, Decision Tree, Naive Bayes, Logistic Regression, and a linear Support Vector Machine. Random Forest [20] was selected for final deployment because it achieved the best overall balance among accuracy, AUC, recall, F1-score, and practical export to embedded C code.

2.4 Edge Deployment

The final prototype was implemented on the STM32F407 MCU. FFT, mean, RMS, and standard deviation were computed with CMSIS-DSP functions, while skewness and kurtosis were implemented in custom firmware. The trained Random Forest model was converted to C code by *m2cgen* and integrated into the embedded project.

The firmware performs ADC acquisition through DMA, signal buffering, preprocessing, 133-feature extraction, model inference, and deterministic alarm logic by LED. Therefore, the prototype evaluates not only the classification model but also the complete practical software chain.

Table 1: Collection scenarios used in the dataset.

No.	Arc Branch	No-Arc Branch	No.	Arc Branch	No-Arc Branch
1	Electric Kettle	None	17	Electric Kettle	Circular Saw
2	Iron	None	18	Electric Kettle	TV
3	Electric Grill	None	19	Electric Grill	Electric Kettle
4	Induction Cooktop	None	20	Electric Grill	Induction Cooktop
5	LED Bulb	None	21	Electric Grill	Soldering Plate
6	Hot Air Rework Station	None	22	Induction Cooktop	Electric Kettle
7	Circular Saw	None	23	Induction Cooktop	Electric Grill
8	Hair Dryer	None	24	Induction Cooktop	Soldering Plate
9	Rice Cooker	None	25	Soldering Plate	Electric Kettle
10	Electric Fan	None	26	Soldering Plate	Electric Grill
11	TV	None	27	Soldering Plate	Induction Cooktop
12	Refrigerator	None	28	Soldering Plate	TV
13	Soldering Plate	None	29	TV	Electric Kettle
14	Electric Kettle	Electric Grill	30	TV	Iron
15	Electric Kettle	Induction Cooktop	31	TV	Induction Cooktop
16	Electric Kettle	Soldering Plate	32	TV	Soldering Plate

2.5 Evaluation Metrics

The classifier was evaluated with accuracy, recall, precision, and F1-score. AUC was also used during model comparison. On the embedded platform, Flash usage, RAM usage, and execution time were measured. Because arc misses are safety critical, recall and F1-score were treated as key performance indicators.

3 Results

3.1 Overall Model Performance

Table 3 summarizes the benchmark results across candidate models. Tree-based ensemble models gave the strongest overall performance. Random Forest achieved 89.2% accuracy, 0.959 AUC, 90.4% recall, and 86.98% F1-score. Although XGBoost reached a very similar accuracy of 89.3%, Random Forest showed a slightly better balance of AUC, recall, and F1-score, which supported its selection as the final deployment model.

The validation confusion matrix is shown in Figure 2. The model produced 10,893 true negatives and 7,403 true positives, while 1,518 normal windows were predicted as arc and 702 arc windows were missed. This result indicates that the model captures most arc events while still leaving room for better false-alarm control.

For the final validation split, the selected Random Forest model achieved 89.18% accuracy, 91.34% recall, 82.98% precision, and 86.96% F1-score. The recall value is especially important for safety-oriented monitoring because it means that most arc windows were detected correctly. At the same time, the false-positive count suggests that later versions should include temporal alarm filtering or decision smoothing to reduce unnecessary alarms.

3.2 LOSO Performance

Figure 3 presents the LOSO accuracy and F1-score for all 32 held-out scenarios. Across all folds, the mean accuracy reached 86.53% and the mean F1-score reached 83.75%, while the median F1-score was 87.41%. Twelve scenarios achieved F1-score of at least 90%, which shows that the model can generalize well in many household conditions. Strong cases include Electric Kettle, Electric Grill, Rice Cooker, Electric Fan, Refrigerator, Soldering Plate, Electric Kettle (Arc) / Soldering Plate (No-Arc), Electric

Table 2: Extracted features used for classification.

#	Feature Name	Formula	Count
1	μ	$\frac{1}{N}\sum_{i=1}^{N} x_i$	1
2	RMS	$\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$	1
3	σ	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}$	1
4	Skewness	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^3}{\sigma^3}$	1
5	Kurtosis	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^4}{\sigma^4}-3$	1
6	FFT	magnitude bins	128
	Total:		**133**

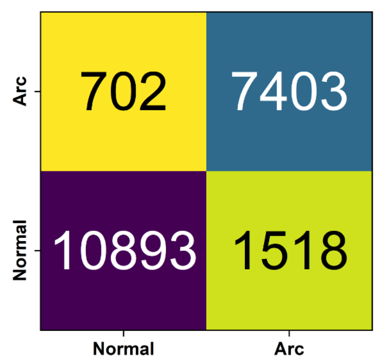

Figure 2: Confusion matrix of the final validation result.

Table 3: Benchmark performance of candidate classifiers.

Model ID	Classifier	Accuracy	AUC	Recall	Prec.	F1-Score
xgboost	Extreme Gradient Boosting	**89.3**	0.958	88.9	**84.7**	86.8
rf	**Random Forest Classifier**	89.2	**0.959**	90.4	83.7	**86.9**
et	Extra Trees Classifier	88.9	0.957	**91.1**	82.6	86.7
gbc	Gradient Boosting Classifier	88.4	0.952	89.8	82.5	86.0
lda	Linear Discriminant Analysis	85.3	0.929	78.2	83.6	80.8
ridge	Ridge Classifier	85.3	0.929	78.0	83.6	80.7
dt	Decision Tree Classifier	84.7	0.839	80.3	80.8	80.5
nb	Naive Bayes	84.0	0.904	78.5	80.5	79.5
lr	Logistic Regression	82.5	0.874	82.3	75.7	78.8
svm	SVM with Linear Kernel	67.7	0.842	84.6	58.8	67.4

Kettle (Arc) / Circular Saw (No-Arc), Electric Grill (Arc) / Electric Kettle (No-Arc), Electric Grill (Arc) / Induction Cooktop (No-Arc), Soldering Plate (Arc) / Electric Grill (No-Arc), and Soldering Plate (Arc) / Induction Cooktop (No-Arc).

The lowest F1-scores were observed in TV (Arc) / Electric Kettle (No-Arc) at 58.83%, TV (Arc) / Soldering Plate (No-Arc) at 60.33%, Induction Cooktop (Arc) / Soldering Plate (No-Arc) at 63.49%, TV (Arc) / Iron (No-Arc) at 65.83%, and Induction Cooktop (Arc) / Electric Kettle (No-Arc) at 68.14%. These results are important because they reveal the limits of a compact feature-based model under hard masking conditions.

One major reason for the low F1-score is the imbalance of branch influence in two-branch scenarios. In several difficult folds, especially the TV arc cases mixed with electric kettle, iron, or soldering plate, the arc branch had a weaker effective signature than the companion load. As a result, the overall current waveform in each window was dominated by the larger no-arc appliance. Because the proposed feature vector summarizes the full window through global statistics and FFT magnitudes, the weak arc disturbance can become only a small perturbation inside a much stronger current pattern. This masking effect reduces separability and makes both false negatives and false positives more likely.

Other causes are also possible. First, TV and induction cooktop loads often contain switching electronics, nonlinear current draw, and time-varying operating modes. Their normal waveforms already include harmonics and irregular behavior that may overlap with arc characteristics. Second, LOSO is a strict protocol. The model does not see the held-out appliance combination during training, so any scenario-specific spectral shape shift directly hurts transfer performance. Third, the 133-dimensional feature set is intentionally lightweight. While this is good for embedded use, it may miss finer transient details that deeper models try to learn directly from waveform or image-like representations [10, 11]. Fourth, the fixed decision boundary of the final classifier is global across devices, so it may not be equally well calibrated for low-power and high-power branches. These observations suggest that future improvement should focus on hard mixed-load cases, especially when a low-power arc source is buried under a high-power non-arc branch.

3.3 Prototype Testing

Prototype testing was used to verify the full practical chain rather than only the offline model. The embedded software includes DMA-based ADC acquisition, signal buffering, preprocessing, 133-feature extraction, Random Forest inference, and LED-based warning logic. The successful integration of both the software pipeline and the trained model on the STM32F407 board demonstrates that the method is feasible for a real embedded device.

The deployed firmware required 896 KB Flash and 102 KB RAM. The total processing time was 7.02 ms per cycle, including 3.25 ms for manual statistical features, 3.53 ms for FFT computation, and 0.23 ms for model inference. Compared with the approximate 200 ms hop interval used for online updates, the processing time provides a wide real-time margin. This means the system still has room for additional functions such as alarm confirmation, communication, or logging.

A running demonstration of the prototype is provided in the video evidence link: https://shorturl.at/vWd6G.

4 Discussion

The results show that a lightweight feature-based method remains competitive for practical household arc detection. The model does not aim to outperform every deep architecture in the literature. Instead, it aims to keep a strong balance between acceptable recognition accuracy, interpretable features, and low embedded cost. This practical balance is important because several recent studies reach very high accuracy with lightweight CNNs, transformers, or other neural models [10, 12], but they often use different datasets, higher sampling rates, or different deployment targets. Therefore, direct numerical comparison is not fully fair.

Another important finding is the value of the two-branch dataset design. If only single-load cases were evaluated, the method would appear stronger than it really is. The mixed-load scenarios reveal the masking problem clearly and show where improvement is still needed. In

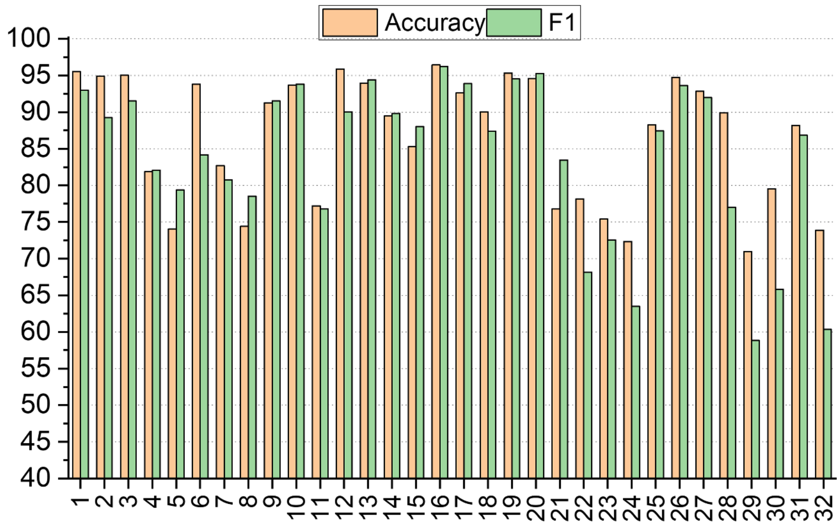

Figure 3: Leave-one-scenario-out performance across the 32 household scenarios.

that sense, the LOSO results are not only an evaluation metric but also a diagnostic tool for dataset design and future model development.

The findings of this study are consistent with earlier work showing that multi-feature current analysis can remain effective when load conditions are diverse and masking effects are present [4, 7, 19]. At the same time, recent deep-learning studies report very high offline accuracy by using CNNs, transformers, or image-like signal representations [9–11]. The present work contributes from a different angle. It shows that a compact Random Forest pipeline can still be valuable when deterministic latency, simple model export, and modest hardware cost are central design requirements. The measured execution time of 7.02 ms on STM32F407 confirms that practical embedded monitoring can be achieved without a more powerful processor, which is an important implication for low-cost protective devices intended for real deployment.

This study has several limitations. First, the dataset is still limited to the collected household scenarios and does not cover all practical operating conditions such as voltage variation, aging appliances, or long-term field noise. Second, the final system uses only current sensing. Additional voltage or event-level information may help in difficult nonlinear loads [19]. Third, the manual feature set was intentionally compact for embedded execution, but this also limits the model's ability to represent subtle transient structures.

Future research will enhance the proposed detection system's robustness and reliability by focusing on several key areas. We will expand the dataset to include more challenging two-branch combinations, targeting scenarios where high-power non-arc loads obscure low-power arc signatures. To address the 1,518 false-positive windows identified, we will implement a temporal filtering strategy or sliding-window voting mechanism to better differentiate between non-arc transients and actual arcing events, reducing nuisance tripping while maintaining the current 91.34% recall rate. We will also explore feature normalization by load power, adaptive thresholds, and probability calibration to refine the decision boundary. Lastly, we aim to develop a hybrid architecture in which a lightweight model serves as a fast first-stage detector, activating a more selective second-stage model only for ambiguous signals.

5 Conclusions

This paper presents the development of a robust, practical, real-time embedded system for AC series arc-fault detection in residential environments. By integrating a specialized two-branch household dataset with a lightweight 133-dimensional feature set and a Random Forest classifier, the proposed system achieved a high validation accuracy of 89.18% and an F1-score of 86.96%. The experimental results from the LOSO evaluation successfully demonstrated the model's generalization capabilities while systematically exposing the masking effects inherent in mixed-load conditions, particularly where high-power appliances obscure low-power arc signatures. Crucially, the end-to-end feasibility was verified through real-time deployment on an STM32F407 microcontroller, where the entire processing pipeline, including feature extraction and inference, was completed in just 7.02 ms per cycle. This significant real-time margin, combined with the moderate hardware resource requirements, confirms that the proposed approach is well-suited for low-cost, edge-based protective devices. Future efforts will further focus on enhancing robustness against masking effects through adaptive thresholding and temporal consistency checks to minimize false alarms in complex electrical networks.

References

[1] Saleh, S. A., Valdes, M. E., Mardegan, C. S., Alsayid, B.: The State-of-the-Art Methods for Digital Detection and Identification of Arcing Current Faults. IEEE Transactions on Industry Applications

55(5) (2019), 4536–4550. https://doi.org/10.1109/TIA.2019.2923764
[2] Lezama, J., Schweitzer, P., Tisserand, E., Humbert, J.-B., Weber, S., Joyeux, P.: An embedded system for AC series arc detection by inter-period correlations of current. Electric Power Systems Research **129** (2015), 227–234. https://doi.org/10.1016/j.epsr.2015.08.005
[3] Tisserand, E., Lezama, J., Schweitzer, P., Berviller, Y.: Series arcing detection by algebraic derivative of the current. Electric Power Systems Research **119** (2015), 91–99. https://doi.org/10.1016/j.epsr.2014.09.011
[4] Vu, H. D., Calderon, E. M., Schweitzer, P., Weber, S., Britsch, N.: Multi criteria series arc fault detection based on supervised feature selection. International Journal of Electrical Power & Energy Systems **113** (2019), 23–34. https://doi.org/10.1016/j.ijepes.2019.05.012
[5] Jiang, J., Li, W., Wen, Z., Bie, Y., Schwarz, H., Zhang, C.: Series Arc Fault Detection Based on Random Forest and Deep Neural Network. IEEE Sensors Journal **21**(15) (2021), 17171–17179. https://doi.org/10.1109/JSEN.2021.3082294
[6] Zhao, H., Liu, J., Lou, J.: Series arc fault detection based on current fluctuation and zero-current features. Electric Power Systems Research **202** (2022), 107626. https://doi.org/10.1016/j.epsr.2021.107626
[7] Li, Z., Liu, Y., Liang, Z., Huang, Y.: A Low Voltage AC Fault Arc Detection Method Based on Multi-Feature Fusion. Journal of Physics: Conference Series **2814**(1) (2024), 012030. https://doi.org/10.1088/1742-6596/2814/1/012030
[8] Yin, Z., Wang, L., Zhang, Y., Gao, Y.: A Novel Arc Fault Detection Method Integrated Random Forest, Improved Multi-scale Permutation Entropy and Wavelet Packet Transform. Electronics **8**(4) (2019), 396. https://doi.org/10.3390/electronics8040396
[9] Chabert, A., Bakkay, M. C., Schweitzer, P., Weber, S., Andrea, J.: A Transformer Neural Network For AC series arc-fault detection. Engineering Applications of Artificial Intelligence **125** (2023), 106651. https://doi.org/10.1016/j.engappai.2023.106651
[10] Wang, Z., Tian, S., Gao, H., Han, C., Guo, F.: An On-Line Detection Method and Device of Series Arc Fault Based on Lightweight CNN. IEEE Transactions on Industrial Informatics **19**(10) (2023), 9991–10003. https://doi.org/10.1109/TII.2023.3233967
[11] Paul, K. C., Chen, C., Wang, Y., Zhao, T.: LArcNet: Lightweight Neural Network for Real-Time Series AC Arc Fault Detection. IEEE Open Journal of Industry Applications **6** (2025), 79–92. https://doi.org/10.1109/OJIA.2024.3522364
[12] Ning, X., Sheng, D., Zhou, J., Liu, Y., Wang, Y., Zhang, H., Lei, X.: Arc_EffNet: A Novel Series Arc Fault Detection Method Based on Lightweight Neural Network. Electronics **12**(22) (2023), 4617. https://doi.org/10.3390/electronics12224617
[13] Tian, C., Xu, Z., Wang, L., Liu, Y.: Arc Fault Detection Using Artificial Intelligence: Challenges and Benefits. Mathematical Biosciences and Engineering **20**(7) (2023), 12404–12432. https://doi.org/10.3934/mbe.2023552
[14] Chu, R., Schweitzer, P., Zhang, R.: Series AC Arc Fault Detection Method Based on High-Frequency Coupling Sensor and Convolution Neural Network. Sensors **20**(17) (2020), 4910. https://doi.org/10.3390/s20174910
[15] Velasquez, J. D., Cadavid, L., Franco, C. J.: Emerging Trends and Strategic Opportunities in Tiny Machine Learning: A Comprehensive Thematic Analysis. Neurocomputing **648** (2025), 130746. https://doi.org/10.1016/j.neucom.2025.130746
[16] Riba, J.-R., Moreno-Eguilaz, M., Ortega, J. A.: Arc Fault Protections for Aeronautic Applications: A Review Identifying the Effects, Detection Methods, Current Progress, Limitations, Future Challenges, and Research Needs. IEEE Transactions on Instrumentation and Measurement **71** (2022), 3504814. https://doi.org/10.1109/TIM.2022.3141832
[17] Choi, W.-K., Kim, S.-H., Bae, J.-H.: Series-arc-fault Diagnosis Using Feature Fusion-based Deep Learning Model. ETRI Journal (2024), etrij.2023-0457. https://doi.org/10.4218/etrij.2023-0457
[18] Mao, Y., Safa, S., Smith, G., Wurth, L., Weiss, R., Hagemeyer, J.: Why AI: A Comparative Study for Detection Methods in DC Series Arc Fault. IEEE Access **13** (2025), 42703–42722. https://doi.org/10.1109/ACCESS.2025.3548309
[19] Zhang, Y., Chen, H. C., Li, Z., Jia, C., Du, Y., Zhang, K., Wei, H.: Lightweight AC Arc Fault Detection Method by Integration of Event-Based Load Classification. IEEE Transactions on Industrial Electronics **71**(4) (2024), 4130–4140. https://doi.org/10.1109/TIE.2023.3277125
[20] Breiman, L.: Random forests. Machine Learning **45**(1) (2001), 5–32. https://doi.org/10.1023/A:1010933404324