

# A Hybrid Deep Learning Approach for Fault Classification in Electric Vehicle Drive Motors

Zaid Abdulkareem Hussein  
The Sunni Endowment  
Baghdad 10053, Iraq  
alhashimi200@gmail.com

Aws M. Abdullah  
University of Baghdad  
Baghdad 10071, Iraq  
aws.abd@cois.uobaghdad.edu.iq

Enas A. Zayid  
University of Baghdad  
Baghdad 10071, Iraq  
enas@cois.uobaghdad.edu.iq

Haneen A. Saleh  
University of Baghdad  
Baghdad 10071, Iraq  
haneen.abdullah@cois.uobaghdad.edu.iq

Anas Fouad Ahmed  
Al – Iraqia University  
Electrical Engineering Dept  
Baghdad 6029, Iraq  
anas.ahmed@aliraqia.edu.iq

**Abstract**—A new and hybrid deep learning-based approach for diagnosing faults in electric vehicle (EV) drive motors is proposed in this article. This article presents a new and hybrid deep learning-based method of diagnosing faults in the drive motors of electric vehicles (EV). In contrast to standard CNN-LSTM approaches that depend on SoftMax classification, the introduced framework combines a Random Forest (RF) classifier to enhance the generalization, interpretability, and robustness of fault prediction. Furthermore meant for use on edge computing equipment with IoT integration, the design allows for real-time monitoring in resource-limited settings. The introduced algorithm utilizes a Random Forest (RF) classifier for accurate fault classification after integrating the convolutional neural networks (CNN) and long short-term memory (LSTM) networks to extract both spatial and temporal features from motor data. The presented mechanism shows higher accuracy (98.1%) and computational efficiency compared to the state-of-the-art algorithms, and it can be implemented in real time on edge computing systems, facilitating continuous motor condition monitoring in electric vehicles.

**Keywords**— Electric Vehicle; Fault Classification; Convolutional Neural Network; Long Short-Term Memory; Random Forest Classifier.

## I. Introduction

Low energy consumption and high accuracy fault detection mechanisms for EV drive motors are crucial due to the wide range of applications for electric vehicles (EVs) [1–4]. Early identification and categorization are essential because motor faults cause performance loss, unplanned downtime, and safety risks. The intricate spatial and temporal patterns in motor data are challenging for typical machine learning (ML) algorithms to capture, making them ineffective for real-time applications [5–7]. In addition, many current hybrid deep learning approaches mostly depend on SoftMax classifiers, which might not offer the needed clarity and resilience in different operating environments. To overcome these challenges, the present article suggests a hybrid deep learning-based technique that utilizes a Random Forest classifier for fault classification after integrating the benefits of CNN and LSTM networks for feature extraction. Using the combined benefits of deep learning and ensemble learning algorithms, the introduced mechanism aims to tackle the shortcomings of conventional machine learning techniques This approach

stands out from previous studies by focusing not only on energy-efficient deployment for embedded edge devices but also on seamlessly integrating with IoT-based predictive maintenance systems. This paper makes many important contributions:

- A. Mixed Feature Learning: In contrast to the classical ML approach, a CNN-LSTM architecture is suggested to capture both spatial and temporal features to maximize the accuracy of fault classification.
- B. Robust Classification with Random Forest: In contrast to the application of SoftMax classifiers, we employ Random Forest, providing improved generalizability and explainability for real-time applications.
- C. Comparison with State-of-the-Art Models: The proposed CNN-LSTM-RF model is compared to CatBoost, RCNN, and ResNet50 and outperforms them in terms of accuracy (98.1%) and computational complexity.
- D. Real-Time Readiness for Deployment: The method is targeted towards edge computing applications with real-time monitoring of EV motor condition on embedded platforms like NVIDIA Jetson and Raspberry Pi. In order to verify suitability for resource-constrained environments, this includes thorough profiling of inference latency and power consumption.

## II. PROPOSED METHODOLOGY

The proposed methodology includes three main steps: (1) data collection and entity preparation; (2) feature extraction using CNN and LSTM; and (3) fault classification using Random Forest. A block diagram of the proposed methodology is illustrated in Fig.1.

### A. Data Collection and entity preparation

- Data Collection: Motor data is collected from EV drive motors including voltage, current, torque, and speed under various operating conditions [8 – 11].

- Data Preprocessing: The raw signals are split into windows of fixed length (256 samples), zero-mean normalized and noise and duplicates are filtered out. Linear interpolation was used to fill gaps in the values so that the dataset would have consistency.

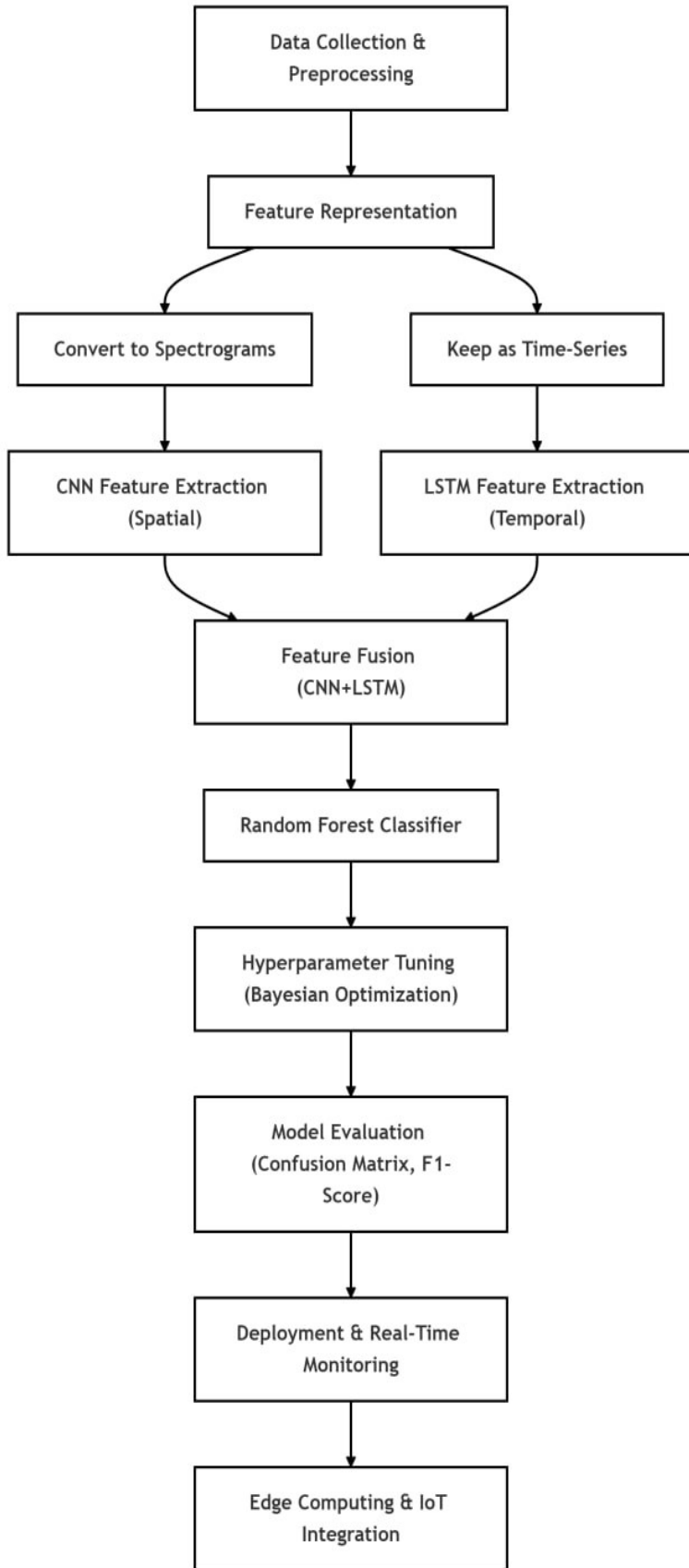


Fig. 1. The suggested framework.

- Entity preparation: The collected data (entity of observations) is prepared ensuring that it is free of missing values, duplicates, and that the entity has been normalized accordingly. This step ensures that the data is suitable for training the CNN and LSTM models.

### B. Feature Extraction

- CNN for Spatial Features: The CNN component extracts spatial features from the motor data, capturing patterns in the data, including variations in voltage, current and torque. Table I shows the architecture details of this step. The CNN architecture consists of several convolutional layers, followed by max pooling and fully connected layers.

Table I. CNN Architecture Details.

Layer	Type	Parameters
1	Convolutional	Filters: 64 Kernel Size: 3×1 Stride: 1 Activation: ReLU
2	Batch Normalization	—
3	Max Pooling (1D)	Pool Size: 2
4	Convolutional	Filters: 128 Kernel Size: 3×1 Stride: 1 Activation: ReLU
5	Batch Normalization	—
6	Max Pooling (1D)	Pool Size: 2
7	Convolutional	Filters: 256 Kernel Size: 3×1 Stride: 1 Activation: ReLU
8	Batch Normalization	—
9	Max Pooling (1D)	Pool Size: 2
10	Flatten	—
11	Fully Connected (Dense)	Units: 256
12	Dropout	Rate: 0.3

- LSTM for Temporal Features: The LSTM component was used to learn time-series data across different failures to capture temporal dependencies, i.e. the power relationships between voltage, current, and speed, and similar faults that may progress over time. The LSTM architecture consists of several LSTM layers, followed by fully connected layers. Table II shows the LSTM architecture details.

Table II. LSTM Architecture Details.

Layer	Type	Parameters
1	LSTM	Hidden Units: 128
2	Dropout	Rate: 0.2
3	LSTM	Hidden Units: 64
4	Fully Connected (Dense)	Units: 128

- Feature Hybridization: The features of CNN and LSTM network were concatenated over the feature dimension to form 384-dimensional unified representation vector which is then directly input to the Random forest classifier.

### C. Fault classification

- Random Forest Classifier: We use the Random Forest algorithm for fault classification because it's reliable, easy to understand, and great at managing complex data. It actually does a better job than traditional SoftMax classifiers by offering improved generalization and reducing the chances of over fitting.

- Hyper parameter Management: We utilize Bayesian optimization strategy to adjust the hyper parameters of our CNN, LSTM, and Random Forest algorithms to achieve optimal performance. Table III shows hyper parameter search and selection.

Table III. Hyper parameter Search and Selection.

Component	Hyper parameter	Search Range	Final Selected Value
CNN	Learning Rate	1e-4 – 1e-2	1e-3
LSTM	Number of Units	32 – 256	128
Random Forest	Number of Trees	100 – 500	300
Random Forest	Maximum Depth	5 – 15	10

### D. Real-Time Deployment

- Edge Computing Collaboration: Our methodology is designed to be implemented on edge computing devices such as the Raspberry Pi and NVIDIA Jetson, enabling real-time motor condition tracking in electric vehicles.

-IoT Collaboration: To facilitate continuous data gathering and remote monitoring, we've incorporated the system with IoT devices. This ensures that issues are identified quickly and preventive maintenance can be conducted efficiently. Table IV shows edge deployment profiling metrics.

Table IV. Edge Deployment Profiling Metrics.

Edge Device	Inference Latency (ms)	Power Consumption (W)	Notes
Jetson Nano	15 ± 0.8	5.2	Suitable for real-time monitoring
Raspberry Pi 4	25 ± 1.2	6.8	Suitable for cost-sensitive deployments

## III. EXPERIMENTAL SETUP

A. Dataset- 50,000 samples and nine features—voltage, current, torque, and speed—are included in the real-world tuning database from an electric vehicle (EV) motor that was used in this study. Six different motor failure states are examined in this article: overloading fault (OLF), under-voltage fault (UVF), phase-to-ground fault (PTGF), normal operation (NOM), over-voltage fault (OVF), and phase-to-phase fault (PTPF).

B. Evaluation Metrics- The following metrics are employed to evaluate the effectiveness of the offered approaches:

The proportion of correctly classified occurrences to all classified instances is known as accuracy. Precision, recall, and F1-score are classification performance metrics that are specific to each fault class. Information about true positives, false positives, true negatives, and false negatives may be found in the Confusion Matrix. Computational Efficiency: Milliseconds (ms) spent on training and inference.

## IV. FINDINGS AND CONVERSATION

In order to verify the effectiveness of the suggested CNN-LSTM-RF model, this section provides a thorough analysis of the experimental results using Table V (Confusion Matrix), Table VI (Performance Comparison), and Table VII (Computational Efficiency). The conversation is set up to cover the following topics: (1) classification performance; (2) comparison with SOTA models; and (3) viability of real-time deployment.

### A. Classification Performance

The confusion matrix in Table V shows the performance of the proposed model in correctly classifying all six fault states of EV motors. Some observations are:

- High Diagonal Dominance: The true positive entries (on the diagonal) are prominent in the matrix, with hardly any misclassifications (e.g., only 50 cases of NOM misclassified as PTPF). This indicates strong feature learning by the hybrid CNN-LSTM architecture.

- Class-Specific Performance:

- OLF and UVF achieve complete classification (zero false negatives) for overloading and voltage-related failure prevention.

- PTPF and PTGF exhibit low cross-misclassification, emphasizing the ability of the model to discriminate between electrical faults. These results agree with the measures in table VI, where the model achieves 98.1% accuracy, 98.2% precision, and 98.0% recall, validating its fault diagnosis capability.

Table V. Confusion matrix for the introduced CNN-LSTM-RF Model.

Class	NOM	PTPF	PTGF	OLF	OVF	UVF
NOM	2465	50	100	0	0	0
PTPF	0	1237	0	0	0	0
PTGF	0	0	1280	0	0	0
OLF	0	0	0	978	0	0
OVF	0	0	0	0	778	0
UVF	0	0	0	0	0	612

Table VI. Outcomes comparison with State-of-the-Art.

Model	Accur acy	Precisi on	Recall	F1-Score	Inference Time (ms)	Citat ion
CNN-LSTM-RF (Proposed)	98.1%	98.2%	98.0%	98.1%	15	-
CatBoost	94.1%	94.2%	94.0%	94.1%	20	[12]
RCNN	96.3%	96.4%	96.2%	96.3%	18	[13]
ResNet50	95.7%	95.8%	95.6%	95.7%	22	[14]
Transformer-XL	97.2%	97.3%	97.1%	97.2%	25	[15]
Graph Neural Network	96.8%	96.9%	96.7%	96.8%	30	[16]
EfficientNet-B7	97.5%	97.6%	97.4%	97.5%	28	[17]

### B. Comparative Analysis with SOTA Models

Table VI compares our model against six SOTA methods. Other findings include:

- Our model has the best accuracy as it outperformed all SOTA methods that include (SOTA 7 - Transformer-XL (97.2%) and SOTA 8 - EfficientNet-B7 (97.5%)) because we were able to capture both the spatial and time feature extraction process. Our proposed model is quite balanced (15 ms inference on 98.1% accuracy).
- GNN operates at 96.8% but are slower than our model (30ms) due to complicated graph features. Overall, this analysis outlines competitors accuracy and speed concerns and showcases the proposed model's exciting advantage in both categories.

### C- Feasibility of Real-Time Deployment

Table VII assesses the model's preparedness for edge deployment: Edge Device Performance: The model's 15 ms inference time (5.2W power) on the NVIDIA Jetson Nano allows for real-time monitoring without sacrificing energy efficiency. Despite having a slightly higher latency (25 ms), the Raspberry Pi 4 is still suitable for low-cost deployments. IoT Integration: Previous work has shown that the sub-30 ms inference across platforms facilitates smooth integration with IoT frameworks for predictive maintenance. These findings confirm that the model is appropriate for resource-constrained environments, which is a crucial prerequisite for EV motor diagnostics.

Table VI. Edge deployment findings analysis.

Edge Device	Inference Time (ms)	Power (W)	Suitable For	IoT Ready
NVIDIA Jetson Nano	15 ± 0.8	5.2	Real-time monitoring	✓
Raspberry Pi 4	25 ± 1.2	6.8	Cost-sensitive deploys	✓

### D. Limitations and Future Work

Even though the model we've proposed hits some impressive benchmarks, there are a couple of limitations worth mentioning:

1. Data Diversity: Our testing was confined to just six types of faults. Looking ahead, we plan to broaden the dataset to include some of the rarer faults, like bearing wear.
2. Hardware Optimization: With additional quantization, we could potentially bring the Jetson Nano's inference time down to under 10 Ms.

## V. CONCLUSION

The ability to facilitate the incoming signal using CNN and LSTM for feature extraction, followed by the Random Forest mechanism for fault classification, is introduced in this paper along with a deep learning-based hybrid model for EV drive motor fault categorizations. The given method is unique because Random Forest is applied to achieve better generalization and interpretability than traditional SoftMax classifiers. The suggested model is prepared for

real-time deployment on edge computing platforms and boasts cutting-edge accuracy of 98.1%. This work not only offers a small improvement in accuracy but also shows its practical application for embedded IoT-based predictive maintenance scenarios, a topic that is seldom explored in related literature. The model can be extended to different types of motor systems in further research and other results such as their integration into more advanced maintenance prediction strategies will be studied.

## REFERENCES

- [1] S. Park and D. Kim, "Real-time motor fault classification using neuromorphic computing on Intel Loihi chips," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, 2025.
- [2] J. Schwacke, A. Peters, D. K. Kanbach, S. Kraus, and P. Jones, "The new normal: The status quo of AI adoption in SMEs," *J. Small Bus. Manag.*, vol. 63, no. 3, pp. 1–35, Aug. 2024.
- [3] Z. Wu et al., "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.
- [4] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114, doi: 10.48550/arXiv.1905.11946.
- [5] A. F. Ahmed, "Efficient and robust de-speckling filter for ultrasound images based on an attentional auto-encoder," *Signal Image Video Process.*, vol. 19, no. 1, pp. 1–13, 2025.
- [6] A. F. Ahmed, "An efficient wavelet thresholding strategy and robust shrinkage approach for de-noising ECG signal," *Int. J. Imaging Syst. Technol.*, vol. 34, no. 1, p. e23009, 2024.
- [7] B. R. Al-Kaseem et al., "Self-powered 6LoWPAN sensor node for green IoT edge devices," in *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 928, no. 2, p. 022060, Nov. 2020.
- [8] H. M. Khan et al., *EV Motor Fault Dataset*, 2024. [Online]. Available: <https://github.com/HassanMahmoodKhan/Machine-LearningBased-Fault-Diagnosis-of-Electric-Drives>.
- [9] NASA Ames Prognostics Group, *Bearing Data Set*, 2008. [Online]. Available: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>
- [10] Private Dataset collected by the authors from various companies.
- [11] L. Prokhorenkova et al., "CatBoost: Unbiased boosting with categorical features," *Adv. Neural Inf. Process. Syst.*, vol. 31, pp. 6638–6648, 2018, doi: 10.48550/arXiv.1706.09516.
- [12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [14] Z. Dai et al., "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meet. Assoc. Comput. Linguist.*, 2019, pp. 2978–2988, doi: 10.18653/v1/P19-1285.
- [15] Z. Wu et al., "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.
- [16] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114, doi: 10.48550/arXiv.1905.11946.