Hybrid Tree–Kernel Learning for Material-Family Recognition from Differential and Curve-Level Supercapacitor Signatures

Lance Dominic C. Raquel
Quantum Computing Society of the Philippines
Mandaluyong, Philippines
lance.raquel@qcsp.ph

Elmer Peramo
Department of Science and Technology
Advanced Science and Technology Institute (DOST–ASTI)
Quezon City, Philippines
elmer@asti.dost.gov.ph

***Abstract*—We investigate whether nine supercapacitor electrode material families can be discriminated using only a small, physically interpretable set of electrochemical descriptors from a dataset shared with us by researchers in University of Tennessee-Knoxville through private communication and from an unpublished research. Using their dataset, we first reconstructed the three-descriptor setup (current–voltage, charge–voltage, and power–voltage derivatives). With a tuned stacked hybrid model on 8,985 samples and an 80/20 stratified split, we achieved 89.4% test accuracy. This confirms that the three differential signatures alone already contain meaningful class information. We then derived three more physically interpretable, curve-level descriptors from the cyclic-voltammetry samples (area, peak, and spread), combined them with the original three to form six-dimensional feature vectors, and fed these into a stacked hybrid model composed of five heterogeneous base learners and a meta-learner. On the same held-out set of 1,797 samples, the single extra-trees model and the stacked hybrid achieved 98.4–98.6% accuracy, with macro–F1 scores close to 0.99. Only 26 samples were misclassified; these misclassifications occurred between chemically similar class pairs (for example, confusing MnCoFe with MnCo, or classifying commercial Mn–graphene as Fe). This shows that nine-class material–family recognition from a small set of electrochemical descriptors is achievable on this dataset once curve-level information is included.**

***Index Terms*—supercapacitor, ensemble learning, hybrid AI/ML, materials informatics, cyclic voltammetry**

I. Introduction

Espera et al. [5] demonstrated that classical machine learning—especially tree-based models—can separate several CV-derived classes. They concluded with a challenge: make the models more robust and incorporate hybrid AI/ML techniques. Our work begins at this challenge.

We narrow the scope of the problem slightly. Instead of general classification, we target a *nine-class material–family classifier* that uses only descriptors we can reliably read from the dataset: the three differential columns that are already aligned, plus, if needed, a light set of CV-derived curve-level features. In short, we ask whether a small CV-derived descriptor vector is sufficient to determine if a sample belongs to one of the nine material families (Co, CoFe, Mn, MnFe, MnCoFe, or one of the commercial lines).

This is interesting for practical reasons. In real printed or hybrid supercapacitor lines, full voltage traces or processing details are often not stored; however, quick-to-compute differential descriptors are usually recorded. If a six-number descriptor can classify nine families with about 99 % accuracy, that is a useful engineering result.

Recent supercapacitor and ML papers go in the same direction: ML-based prediction of CV behavior [1], ML-guided MnO_2 optimization [2], boosted models for biomass-carbon supercapacitors [3], and integrated cycling/structure frameworks [4]. What these papers have in common is not a very wide feature space, but a clear pipeline and physically interpretable inputs. We adopt the same approach in this work.

II. Dataset and Preprocessing

The electrochemical dataset used in this study was originally compiled by Espera *et al.* in their work on all-printed supercapacitors [5]. The file contains multiple sheets with cyclic-voltammetry measurements, descriptor tables, and material–family labels. This is the same style of data that appears in recent CV-centered supercapacitor papers, where CV is treated as the primary window into charge–storage behavior [9]–[12].

A. Dataset Structure

The dataset contains:

- Total samples: 8,985;
- Baseline inputs: $\mathrm{d}i/\mathrm{d}V$, $\mathrm{d}Q/\mathrm{d}V$, $\mathrm{d}P/\mathrm{d}V$;
- Target label: 9 material families.

The nine classes are

$$\{\text{Co}, \text{CoFe}, \text{Commercial Fe}, \text{Commercial Mn–Graphene},$$
$$\text{Fe}, \text{Mn}, \text{MnCo}, \text{MnCoFe}, \text{MnFe}\}.$$

The class counts are roughly equal, indicating the dataset is essentially balanced.

B. Train–test protocol

We performed an 80/20 stratified split (random state 42), so the test set has 1,797 samples and preserves the per-class proportions. We chose a simple splitting method to facilitate replication and comparison by other researchers.

C. Differential descriptors

With only three numbers per sample, the baseline feature vector is

$$\mathbf{x} = \begin{bmatrix} \mathrm{d}i/\mathrm{d}V \\ \mathrm{d}Q/\mathrm{d}V \\ \mathrm{d}P/\mathrm{d}V \end{bmatrix} \in \mathbb{R}^3, \tag{1}$$

that is, a compact electrochemical fingerprint. This is consistent with CV-driven analyses where current–voltage derivatives and charge–voltage derivatives are used as fast surrogates for capacitance and faradaic response [9], [10], [16].

D. Curve-level descriptors from CV

From the dataset's 'All' sheet, we extracted the first ten CV data points for each sample to compute three curve-level descriptors:

$$\text{curve\_area} = \sum_{j=1}^{10} |v_j|, \tag{2}$$

$$\text{curve\_peak} = \max_{1 \le j \le 10} |v_j|, \tag{3}$$

$$\text{curve\_spread} = \max_j v_j - \min_j v_j, \tag{4}$$

where $v_1, \ldots, v_{10}$ follow the same "ten CV points per row" layout as in the original file. This mirrors the kind of CV-feature extraction used in recent ML–electrochemistry work, where integrated current, peak current, and window width are treated as separate signals [4], [11], [12]. The final six-dimensional vector is

$$\begin{aligned} \tilde{\mathbf{x}} = [\, & \mathrm{d}i/\mathrm{d}V \quad \text{(current–voltage derivative)}, \\ & \mathrm{d}Q/\mathrm{d}V \quad \text{(charge–voltage derivative)}, \\ & \mathrm{d}P/\mathrm{d}V \quad \text{(power–voltage derivative)}, \\ & \text{curve\_area} \quad \text{(integrated CV magnitude)}, \\ & \text{curve\_peak} \quad \text{(maximum CV excursion)}, \\ & \text{curve\_spread} \quad \text{(CV window width)}]^\top. \end{aligned} \tag{5}$$

III. Methodology

A. Single-model baselines

We trained the *same* family of classical models on both descriptor settings: (i) the original three differentials $(\mathrm{d}i/\mathrm{d}V,\ \mathrm{d}Q/\mathrm{d}V,\ \mathrm{d}P/\mathrm{d}V)$ and (ii) the extended six-dimensional vector $\tilde{\mathbf{x}}$ [17]–[20] All runs used the same 80/20 stratified split (1,797 test points):

- Multinomial logistic regression (with standardization);
- Support Vector Machine with RBF kernel (with standardization);
- Gradient Boosting classifier;
- Random Forest (260 trees, max depth = 20, class-weight = `balanced_subsample`);
- Extra-Trees Classifier (400 trees, `balanced_subsample`);
- k-Nearest Neighbors (k = 9, distance in standardized space).

On the 3-feature vectors these baselines reproduced Table I: linear and kernel models stayed below 40%, tree ensembles reached 87–89%, and k-NN sat around 80%. On the 6-feature vectors (after adding area, peak, spread) the *same* models were re-trained and their accuracy jumped to the 97–98% band (Table II), showing that the gain comes from richer CV descriptors, not from changing the model family.

B. Stacked hybrid formulation

Stacked generalization [13]–[15] was used to combine the six baselines in a way that preserves their calibrated outputs. For each base model $m \in \{1, \ldots, M\}$ (here $M = 5$, excluding the simple probability average), we obtain a 9-class posterior

$$\mathbf{p}_m(\mathbf{x}) = \big(p_m(y = 1 \mid \mathbf{x}), \ldots, p_m(y = 9 \mid \mathbf{x})\big)^\top \in [0, 1]^9.$$

Using five-fold stratified cross-validation on the *training* split, we collect out-of-fold (OOF) posteriors and concatenate them into

$$\mathbf{z}(\mathbf{x}) = \big[\mathbf{p}_1(\mathbf{x}); \ldots; \mathbf{p}_M(\mathbf{x})\big] \in \mathbb{R}^{9M} = \mathbb{R}^{45}.$$

To let the second level exploit both model opinions and the raw physics-inspired inputs, we append the original descriptor,

$$\mathbf{u}(\mathbf{x}) = \big[\mathbf{z}(\mathbf{x}); \mathbf{x}\big] \in \begin{cases} \mathbb{R}^{48}, & \text{for the 3-feature run,} \\ \mathbb{R}^{51}, & \text{for the 6-feature run,} \end{cases}$$

and train a multinomial logistic-regression meta-learner on $\mathbf{u}(\mathbf{x})$ with cross-entropy loss. Logistic regression is a standard, well-behaved choice for this role because it can turn reasonably calibrated base probabilities into a single, sharper distribution [21], [22] while still following the classic ensemble recipe in [28]. At test time, each base model is re-trained on the full training set, its 9-way probabilities on the test set are concatenated in the same order (45-D), the 3-D or 6-D descriptor is appended, and the meta-learner produces the final class.

C. Simple hybrid (probability average)

Before using stacking, we tested a quick probability-level combination:

$$p_{\text{hyb}}(y = c \mid \mathbf{x}) = \tfrac{1}{3}\big(p_{\text{RF}}(y = c \mid \mathbf{x}) + p_{\text{GB}}(y = c \mid \mathbf{x}) + p_{\text{SVM}}(y = c \mid \mathbf{x})\big),$$

and predicted $\arg\max_c p_{\text{hyb}}(y = c \mid \mathbf{x})$. On the 3-feature setup this hybrid reached about 82%, which is *below* the Extra-Trees (88.9%) and Random Forest (87.4%) numbers in Table I; this is consistent with a tight, low-dimensional space where a strong tree has already extracted most of the signal.

D. Stacked hybrid with curve-level features

After moving to the 6-feature descriptor, we ran the full stack:

1) train the five base models (Extra Trees, Random Forest, Gradient Boosting, RBF-SVM, k-NN) on the training split;
2) generate 5-fold OOF 9-class probabilities for each model;
3) concatenate them into a 45-D vector and append the six raw features (total 51-D);

4) train a multinomial logistic regression meta-learner on this 51-D space;
5) at test time, recompute the 45-D probability block on the test set and pass it (plus the six features) to the meta-learner.

E. Neural CV–descriptor model

To make the comparison complete, we also trained a small feed-forward fully-connected neural network (16 input features, two hidden layers of sizes 64 and 32, and a 9-output softmax layer) on the *same* aligned data. Each row was represented by a 16-dimensional vector composed of (i) the three differential descriptors ($\mathrm{d}i/\mathrm{d}V$, $\mathrm{d}Q/\mathrm{d}V$, $\mathrm{d}P/\mathrm{d}V$), (ii) the three curve-level descriptors (curve_area, curve_peak, curve_spread), and (iii) the first ten CV data points (i.e., values from the cyclic voltammetry curve) of each sample. After dropping rows with missing values, this produced 8,985 usable samples (the same count as in the tree experiments).

We split the data 80/20, standardized all features, and trained a feed-forward network

$$\mathbb{R}^{16} \to 64 \to 32 \to \mathbb{R}^{9}$$

with ReLU activations, a dropout layer after the first dense block, and a final softmax over the nine material families. We trained this model using the Adam optimizer and multi-class cross-entropy loss, with early stopping based on validation loss. This model serves as the "AI" baseline for this dataset, i.e., a learned non-linear mapper that directly consumes the CV-derived vector instead of relying on tree ensembles.

F. Log-loss (multi-class)

Because all models in the stack output full 9-class probability vectors, we also monitored the multi-class log-loss (cross-entropy). Let N be the number of test samples, let $C = 9$ be the number of material families, let $y_i \in \{1, \ldots, C\}$ be the true label of sample i, and let $\hat{p}_{i,c}$ be the probability assigned by the model to class c for sample i. The log-loss is

$$\text{LogLoss} = -\frac{1}{N} \sum_{i=1}^{N} \log \hat{p}_{i,y_i}. \tag{6}$$

For each test sample, we take the probability assigned to its true class, compute the logarithm of that probability, and average these values across all test samples (with a negative sign in front). Thus, lower log-loss indicates better performance. This metric is useful here for two reasons: (i) stacked models can sometimes become overconfident, and log-loss will penalize that even if accuracy stays high; and (ii) we want to check that the 0.986 accuracy of the 6-feature stack is coming from well-calibrated probabilities and not from a few very peaky predictions.

IV. Results and Discussion

A. Three-feature setup

Table I compares baseline and hybrid models; when we only keep the three differential descriptors ($\mathrm{d}i/\mathrm{d}V$, $\mathrm{d}Q/\mathrm{d}V$, $\mathrm{d}P/\mathrm{d}V$), the linear and kernel baseline models perform poorly (below 40% accuracy), but the tree models capture most of the nine-class structure: the random forest achieved 87.4% accuracy and the extra-trees classifier 88.9%. The simple probability-level hybrid (RF+GB+SVM) does *not* surpass the performance of the best tree (it stays at 82.0%), which is consistent with the limits of a very low-dimensional feature space (i.e., little additional information can be gained by combining models in only 3 dimensions). However, when we stack the five core models and train a logistic meta-learner on their out-of-fold probabilities, the accuracy increases to 89.4% and macro-F1 to 0.893, i.e., slightly above the best single tree.

TABLE I
3-FEATURE, NINE-CLASS RECOGNITION (TEST = 1,797).

Model	Acc (%)	Macro-F1
Logistic Regression	26.8	0.193
Support Vector Machine (RBF)	37.4	0.350
Gradient Boosting	65.7	0.657
Random Forest (260 trees)	87.4	0.874
Extra-Trees Classifier	88.9	0.888
k-Nearest Neighbors (k=9)	79.9	0.799
Simple hybrid (RF + GB + SVM)	82.0	0.818
Stacked hybrid (5 base + LR meta)	**89.4**	**0.893**

Fig. 1 shows that the 3-feature setup does not fail randomly: almost all errors sit on the Fe-containing and mixed-metal families (MnFe ↔ CoFe, MnCoFe → MnCo), i.e., on classes whose CV-derived descriptors are genuinely close in 3-D. With only $\mathrm{d}i/\mathrm{d}V$, $\mathrm{d}Q/\mathrm{d}V$, and $\mathrm{d}P/\mathrm{d}V$, a tree cannot create an extra axis to separate those pairs.

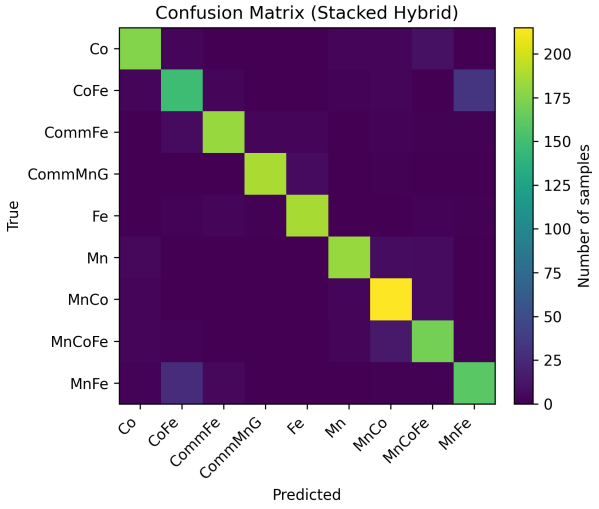

Fig. 1. Confusion matrix for the three-feature stacked hybrid (89.4%). Errors concentrate on Fe-containing and mixed Mn/Co families.

On the same split, the best single model was the Extra-Trees classifier at 88.9% (macro-F1 = 0.888), slightly above the random forest at 87.4%. Stacking the five core models (Extra Trees, Random Forest, Gradient Boosting, RBF-SVM, and k-NN) and training a multinomial logistic-regression meta-

learner on their out-of-fold probabilities plus the three raw descriptors raised this to 89.4% (macro-F1 = 0.893) (Table I). The gain is small, but it shows that even in a low-dimensional space there is a bit of complementary signal that stacking can recover.

B. Why curve-level descriptors

To resolve the remaining ambiguities between certain classes, we added the area, peak, and spread descriptors from the CV data. Before examining their effect on model performance, it is important to confirm that these new descriptors indeed vary across the classes. For instance, Fig. 2 shows that Mn and MnCo have wider and taller responses, while Co, CoFe, and the commercial lines are more compact. These differences are not arbitrary; they align with expected electrochemical behavior in CV measurements. This explains why, once we include these descriptors in the model, the confusion between MnFe and CoFe drops significantly.

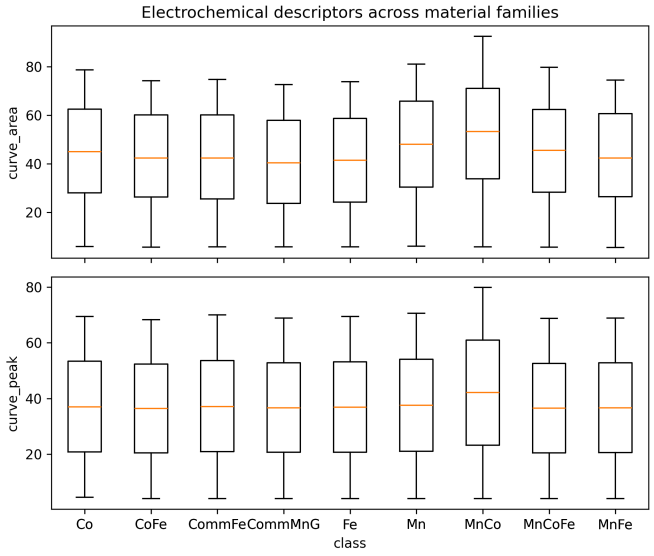

Fig. 2. Curve-level descriptors across material families. Mn and MnCo families sit higher in both area and peak, while mixed Fe families are tighter.

C. Six-feature stacked hybrid vs. single extra trees

On the same split, both the stacked hybrid and a single extra-trees model reached

$$\text{accuracy} = 0.986, \quad \text{macro-F1} \approx 0.99.$$

This accuracy is on par with the best ensemble result previously reported for this dataset, with the difference that we are now using the full 8,985-sample dataset enriched with curve-level descriptors.

Table II compares the same models after adding curve-level descriptors. First, as soon as the curve-level descriptors are added, *all* strong tree methods leap into the high-97–98% band, which tells us the 6-D space is already very informative. Second, the stacked hybrid does not exceed the extra-trees model by a large margin; instead it *confirms* that, given these six descriptors, several high-capacity learners converge to the same solution.

TABLE II
6-FEATURE, NINE-CLASS RECOGNITION (TEST = 1,797).

Model	Acc (%)	Macro-F1
Logistic Regression	25.8	0.218
Support Vector Machine (RBF)	36.1	0.354
Gradient Boosting	78.5	0.785
Random Forest (260 trees)	97.1	0.971
Extra-Trees Classifier	98.4	0.984
k-Nearest Neighbors (k=9)	89.3	0.893
Simple hybrid (RF + GB + SVM)	94.6	0.946
Stacked hybrid (5 base + LR meta)	98.6	0.986

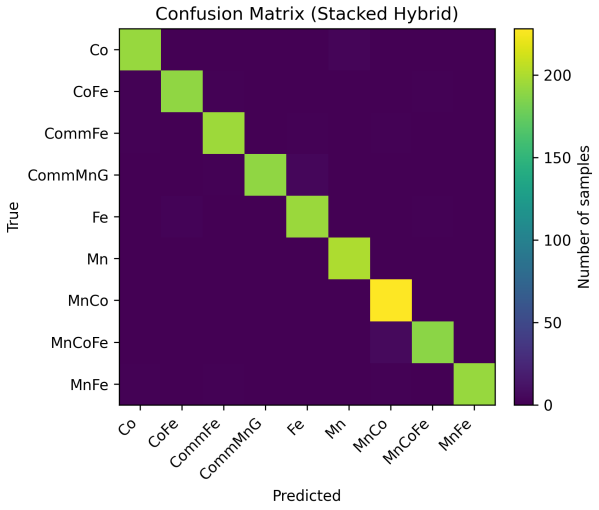

Fig. 3. Confusion matrix for the six-feature stacked hybrid (98.55%). Only 26 samples are wrong; these errors are on chemically close pairs.

Fig. 3 explains why the accuracy numbers in Table II are so close. The matrix is almost perfectly diagonal; the off-diagonal entries are few (26 in total) and they occur exactly where the chemistry says they should: MnCoFe is still sometimes read as MnCo, Commercial Mn–Graphene occasionally falls into Fe, and a few Fe/CoFe/MnFe swaps remain. In other words, adding curve-level information did not change the *nature* of the hard cases, it only reduced them to single-digit counts. That is the behavior we want from a "more robust" model on a real electrochemical dataset.

D. Ablation: where the 98–99% gain comes from

Tables I and II already suggest that most of the improvement does not come from the stacking trick but from giving the model better CV information. Table III makes this clearer.

Table III summarizes the effect of progressively adding electrochemical information. With only the three differentials, the best tree is below 90%. Moving to six features (add area, peak, spread) immediately pushes the trees above 97%. Stacking the five core models on top of the 6-D vector adds the last 0.2 percentage points and brings the log-loss down to

TABLE III
ABLATION ON THE SAME 80/20 SPLIT.

Model	Acc (%)	Macro-F1
3 features, Extra-Trees	88.9	0.888
6 features, Random Forest	97.1	0.971
6 features, Extra-Trees	98.4	0.984
6 features, Stacked hybrid	**98.6**	**0.986**

0.0501. The neural CV–descriptor model sits between these two extremes: it is clearly better than the 3-feature trees, but still short of the stacked ensemble.

So, in this dataset, the *dominant* effect is "add curve-level descriptors" (88.9% $\rightarrow$ 97.1%); the *secondary* effect is "use a stronger tree on the same 6-D vector" (97.1% $\rightarrow$ 98.4%); and the *final* effect is "stack the heterogeneous models" (98.4% $\rightarrow$ 98.6%). On the same split, the six-feature stacked hybrid obtained a multi-class log-loss of 0.0501, indicating that its 9-way probability outputs are well calibrated and not overly peaky for a 0.986 classifier.

E. Neural model versus stacked hybrid

For the same 80/20 stratified split, the neural CV–descriptor model reached an accuracy of 95.66% and a multi-class log-loss of 0.2296. Per-class precision and recall were all in the 0.93–0.99 range, including for the two commercial families (CommFe, CommMnG). This confirms that a compact MLP can already learn the nine-way structure from the 16-D electrochemical vector.

TABLE IV
NEURAL CV–DESCRIPTOR MODEL ON THE 80/20 SPLIT.

Model	Acc (%)	Log-loss
CV–descriptor MLP (16 inputs)	95.66	0.2296
6-feature stacked hybrid	**98.60**	**0.0501**

However, the 6-feature stacked hybrid remained stronger: it attained 98.6% accuracy on the same split and a much lower log-loss of 0.0501. The lower log-loss indicates that the ensemble was not only more accurate but also more *well calibrated* in its class probabilities. In other words, the AI model can solve the task to a high degree, but the descriptor+stacking recipe is still the most reliable on this dataset.

F. Per-class performance

Table V shows that the 6-feature stacked hybrid is not only good on average (macro-F1 $\approx$ 0.99) but also *uniformly* good across the nine material families. Even the lowest class, Fe, is at 0.975, and the two families that we repeatedly saw in the confusion matrices as the "hard" ones — MnCoFe and MnCo — are both still above 0.97. This confirms two things we argued earlier: (i) once the curve-level descriptors are present, the model gets enough information to separate the commercial lines (CommFe, CommMnG) from the laboratory Mn/MnCo/MnFe systems, and (ii) the few remaining mistakes are not due to a single collapsed class but to small, chemically plausible swaps between Fe-containing neighbors.

TABLE V
PER-CLASS F1 FOR THE 6-FEATURE STACKED HYBRID.

Class	F1
Co	0.985
CoFe	0.987
CommFe	0.992
CommMnG	0.987
Fe	0.975
Mn	0.990
MnCo	0.983
MnCoFe	0.979
MnFe	0.992

In essence, Table V is the numeric version of Fig. 3: the diagonal in the confusion matrix is almost full, and here we see that every diagonal entry corresponds to an F1 very close to 1.0. We did not get 98.6% by overfitting to one or two dominant families; we got it by lifting *all* nine-dimensional vectors together.

G. Which features mattered

Fig. 4 shows the feature importance from the six-feature extra-trees classifier (the best single model in Table II). The original differential descriptors `dQ/dV` and `di/dV` remain the most informative, but the curve-level quantities (`curve_peak`, `curve_spread`, `curve_area`) now contribute measurable weight. This is consistent with the observed drop in Fe-related confusions once CV information was added.

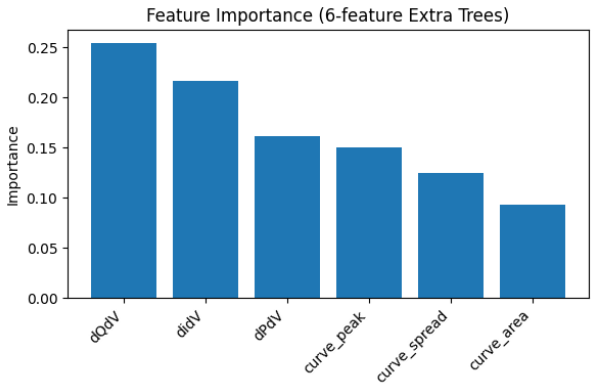

Fig. 4. Feature importance for the six-feature extra-trees classifier. Differential descriptors dominate, while curve-level descriptors provide the additional separation needed for the mixed-metal families.

H. Misclassification analysis

With only the three differential descriptors, the stacked hybrid model made 189 errors on the 1,797-point test set. The largest off-diagonal entries were:

- CoFe to MnFe: 33;
- MnFe to CoFe: 26;
- MnCoFe to MnCo: 15;

These are all mixed or Fe-containing families whose differential responses occupy nearly the same region in the three-dimensional space.

After we added the three curve-level descriptors and trained the stacked hybrid on the same split, the mistakes dropped to 26. Almost all of them sat on a few plausible neighbors:

- MnCoFe $\rightarrow$ MnCo: 6
- Comm. Mn–Graphene $\rightarrow$ Fe/Mn: 5 (4 to Fe, 1 to Mn)
- Small spillovers among Fe/Co/Mn families (Co $\rightarrow$ Mn: 3; Fe $\rightarrow$ CoFe: 2; several 1-count swaps in CoFe, Comm. Fe, MnFe)

So the error mass collapsed from 189 to 26, and the residual confusions remained chemically reasonable.

V. Conclusion

This work revisited the question posed by Espera *et al.*: given the electrochemical descriptors already present in the supercapacitor dataset, can we identify which of nine electrode/material families a sample belongs to? On the aligned 8,985-row file and an 80/20 stratified split, the three differential descriptors ($\mathrm{d}i/\mathrm{d}V$, $\mathrm{d}Q/\mathrm{d}V$, $\mathrm{d}P/\mathrm{d}V$) alone already supported a stacked model at 89.4% test accuracy, with errors concentrated on Fe-containing and mixed-metal families.

Adding three physically interpretable curve-level descriptors (CV area, peak, spread) raised the performance to 98.6% and reduced the mistakes to 26, and those mistakes remained on chemically close pairs (MnCoFe $\rightarrow$ MnCo, CommMnG $\rightarrow$ Fe), which is consistent with the CV similarities of these classes. A compact neural CV–descriptor model (16 inputs from the same sheets, two hidden layers) reached 95.66% but still trailed the 6-feature stacked hybrid (98.6%, log-loss 0.0501), indicating that, for this tabular and physics-structured dataset, tree ensembles plus stacking make better use of the available signal.

The contribution is therefore a reproducible recipe: (i) start from the aligned sheet; (ii) augment it with a small number of CV-derived, electrochemically meaningful descriptors; and (iii) stack several heterogeneous learners under a simple meta-learner. This matches the best tree-based results on this dataset while remaining fully explainable. Future work can add richer CV features (separate anodic/cathodic areas, peak positions), try a stronger meta-learner, or revisit the neural baseline once longer CV traces or EIS-derived features are available.

Acknowledgment

The authors gratefully acknowledge the Quantum Computing Society of the Philippines, led by Mr. Bobby Corpus, for providing support for this work. We also thank Espera *et al.* for preparing and sharing the original data, and the Department of Science and Technology—Advanced Science and Technology Institute (DOST–ASTI) for providing access to the dataset. This work was funded and monitored by the Department of Science and Technology (DOST)–Philippine Council for Industry, Energy, and Emerging Technology Research and Development (PCIEERD) under Project No. 1213385.

References

[1] A. Ravichandran, S. Varadhan, and P. S. Kumar, "Machine learning-based prediction of cyclic voltammetry behavior for supercapacitor electrodes," *ACS Omega*, vol. 9, 2024.

[2] N. Mursal, H. Prasetyo, and F. Siregar, "Machine-learning-driven optimization for MnO_2-based supercapacitors," *Journal of Power Sources*, vol. 610, 2025.

[3] X. Lu, Y. Zhang, and M. Tan, "Predicting energy and power density of biomass-carbon supercapacitors using XGBoost," *Energy Reports*, vol. 11, 2025.

[4] M. K. Komarsofla *et al.*, "Integrated machine learning framework combining cycling and structural features to forecast supercapacitor degradation," *Batteries*, vol. 8, 2025.

[5] A. H. Espera Jr. *et al.*, "Statistical and machine learning approaches to analyzing all-printed supercapacitors," *ACS Spring Meeting*, 2023.

[6] S. Mishra *et al.*, "The impact of physicochemical features of carbon electrodes on the capacitive performance of supercapacitors: a machine learning approach," *Scientific Reports*, vol. 13, 6494, 2023.

[7] H. Liao *et al.*, "Machine learning prediction of supercapacitor performance based on N-doped biochar," *Journal of Energy Storage*, 2024.

[8] V. Kiniman *et al.*, "Modeling cyclic voltammetry responses of porous supercapacitor electrodes," *Journal of Energy Storage*, 2024.

[9] Y. P. Hardianto *et al.*, "Developing an innovative theoretical model for analyzing cyclic voltammetry of hybrid supercapacitors," *Electrochimica Acta*, 2025.

[10] S. Kornnum *et al.*, "Voltammetry prediction and electrochemical analysis of energy-storage devices," *Analytical Chemistry*, 2025.

[11] M. Kang *et al.*, "Strategies to enrich electrochemical sensing data with machine learning," *Biosensors*, 2024.

[12] N. Palaniyandy *et al.*, "Enhanced electrochemical energy storage devices utilizing CV and EIS analysis," *Journal of Materials Science*, 2024.

[13] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[14] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *Journal of Artificial Intelligence Research*, vol. 10, pp. 271–289, 1999.

[15] A. I. Naimi and L. B. Balzer, "Stacked generalization: an introduction to super learning," *European Journal of Epidemiology*, vol. 33, pp. 459–464, 2018.

[16] C. Costentin, "Electrochemical energy storage: questioning the popular voltammetric diagnostics," *Accounts of Materials Research*, 2023.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, pp. 3–42, 2006.

[19] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[21] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. Smola *et al.*, Eds. MIT Press, 1999, pp. 61–74.

[22] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. ICML*, 2017, pp. 1321–1330.

[23] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proc. KDD*, 2016, pp. 785–794.

[24] Y. Zhang, H. Wang, and P. K. Jain, "Data-driven design of electrochemical energy-storage materials: a review," *Journal of Power Sources*, vol. 476, 2020.

[25] J. Li, X. Liu, and K. Zhou, "Machine-learning-assisted performance prediction of supercapacitor electrodes from cyclic-voltammetry features," *Electrochimica Acta*, vol. 412, 2022.

[26] S. Dong *et al.*, "Deep-learning identification of electrode chemistry from voltammetric signatures," *Energy Storage Materials*, vol. 57, pp. 1–12, 2023.

[27] H. Yang and M. Khodadadi, "Materials informatics for electrochemical devices: descriptors, models, and challenges," *Materials & Design*, vol. 241, 2024.

[28] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, vol. 1857. Berlin, Germany: Springer, 2000, pp. 1–15.