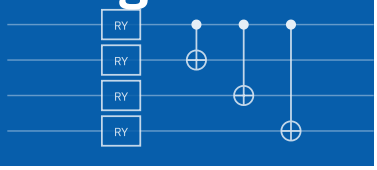
Quantum encoding strategies for electrical grid stability classification

A comparative study under a 10-qubit budget



Ziad Ahmed · Seif Haytham · AbdUlRahman Hany · Abdelrahman Gabr
Team QEUI · Quantum Computing Club, Egypt University of Informatics

code github.com/SEIF-HAYTHAM/QHorizon-2026
page horizon.qeui.gabr.online

SUMMARY

We ask whether a quantum classifier of at most ten qubits can match a tuned classical model at predicting the stability of a simulated four-node power grid, and which design choices decide the answer. On identical data a classical RBF support vector machine beats both our best variational circuit and our best quantum-kernel SVM. The more useful results are the reasons: the three kernel encodings reduce to classical formulas, amplitude encoding discards the sign of the data, entangling gates lowered the kernel scores even after tuning, and one quantity, how far a feature rotates a qubit, moved results more than any circuit change: made trainable, it lifted the variational circuit from 0.560 to 0.836 in cross-validation.

0.902 HOLDOUT MACRO-F1
Classical RBF SVM

0.830 HOLDOUT MACRO-F1
Variational circuit, 4 qubits

0.813 HOLDOUT MACRO-F1
Angle-kernel SVM, 8 qubits

100 SHARED HOLDOUT ROWS
Each finalist scored once

01 Question and dataset

The brief asks: *can a quantum model learn to classify real-world data, and when is it worth it?* We chose a problem whose physical structure a circuit can mirror. The UCI Electrical Grid Stability Simulated Data[1] holds 10,000 simulated operating points of a four-node star grid, one producer and three consumers, under decentralised smart-grid control[2]. Each row gives every node's reaction time (`tau1`–`tau4`), nominal power (`p1`–`p4`) and price-elasticity coefficient (`g1`–`g4`). The label `stabf` marks the grid as stable (3,620 rows) or unstable (6,380 rows). It is public (CC BY 4.0) with no missing values.

Two columns must never reach a model. `stab` is the real part of the characteristic root whose sign defines the label, so using it leaks the answer; the loader checks that sign and label agree on every row, then drops it. `p1` equals −(p2+p3+p4) by power balance (largest deviation 5.8×10^{-15}), so it adds nothing. Eleven candidate inputs remain.

Why eight features. The brief caps circuits at ten qubits. Eight features fit all three encodings without padding: eight qubits for angle encoding, four for dense-angle, three for amplitude ($2^3 = 8$). An ANOVA F-test fitted on training rows only keeps the four reaction times and the four price coefficients. We fixed the count before looking at any score. The data is simulated, from one grid topology: no evidence about real grids.

02 A protocol built for a fair comparison

- **Same rows.** Seed 42 gives a stratified 8,000/2,000 split. Simulating circuits on 8,000 rows is slow, so every matched comparison uses 400 training and 100 holdout rows drawn inside it (stable/unstable 145/255 and 36/64).
- **Same inputs.** The classical control receives exactly the arrays the circuits receive: the eight selected features, standardised and scaled to [0, π] with training-row statistics only. It is the best of six tuned model families, an RBF SVM (C = 10, γ = `scale`).
- **No leakage in tuning.** Three stratified folds of the 400 training rows; feature selection and scaling are refitted inside every fold, and the quantum notebooks reuse the saved fold arrays.
- **One metric, chosen first.** Macro-F1, because 64% of rows are unstable: a majority-class dummy scores 0.64 accuracy but 0.390 macro-F1.
- **Selection by cross-validation only.** Each family's winner is chosen by mean validation macro-F1, refitted on the 400 rows and scored once on the holdout. No holdout score chose a model, an encoding, a feature count or a threshold.
- **One earlier holdout use, disclosed.** Before the encoding comparison, a first kernel SVM with a Z feature map was scored once on the holdout (macro-F1 0.644, accuracy 0.71; cross-validation 0.637; commit 374859a), then replaced. It chose no model reported here.
- **Checkable.** Predictions are saved with row IDs. A validator recomputes every metric from them, and a clean Python 3.13 environment reproduced all saved predictions exactly.

03 Quantum model design, and why

Three encodings that trade qubits for depth

The encoding is the model's only view of the data, so we compare three that spend the qubit budget differently. Angle encoding is shallow and hardware-friendly but needs one qubit per feature. Dense-angle halves the qubits by using two rotation axes per qubit. Amplitude encoding needs only $\log_2 8 = 3$ qubits, but state preparation costs depth and entangling gates, and normalising each row discards its length.

ENCODING	EACH QUBIT CARRIES	QUBITS	DEPTH	GATES	CX
Angle	$RY(x_i)$, one feature	8	4	32	0
Dense-angle	$RY(x_{2q})$ $RZ(x_{2q+1})$	4	4	16	0
Amplitude	$x/\|x\|$ as amplitudes	3	17	23	4

Table 1. Encoding cost for eight features, transpiled to the basis (rz, sx, x, cx) at optimisation level 1 on a demonstration vector.

Quantum-kernel SVM: the encoding in isolation

Each encoding defines a fidelity kernel[3], $K(x, z) = |\langle\varphi(x)|\varphi(z)\rangle|^2$, which we compute exactly from simulated statevectors and pass to a support vector machine with $C = 1$, fixed across encodings. It has no circuit parameters to train, so it measures what the encoding alone can do.

Variational classifier: a circuit shaped like the grid

A trained circuit lets us test design choices one at a time. Ours mirrors the grid (Figure 1): four qubits, one per node, with the producer on qubit 0. Each of three layers re-uploads the node's own data as $RY(s\cdot\tau_i + b)$ and $RZ(s'\cdot g_i + b')$ with trainable scales and shifts, applies trainable RY and RZ rotations, then entangles along the grid's star with CNOTs from the producer to each consumer. Re-uploading[4] lets a four-qubit circuit read eight features three times. The prediction is a weighted sum of $\langle Z\rangle$ on all four qubits passed through a sigmoid: 77 trainable parameters in total.

Training setup

The loss is class-weighted cross-entropy, so the minority stable class counts as much as the majority. We train with Adam on mini-batches of 64 for 120 epochs with a cosine learning rate from 0.05 to 0.01 and exact gradients through the statevector simulator. Rotation angles start near zero ($\sigma = 0.1$). Each fit runs three restarts and keeps the one with the lowest *training* loss, never the best validation score.

An ablation, not an assumption

We did not assume the design was good. Nine variants, trained with the same recipe on the same folds, each change one decision (Table 2). Variant H is a standard eight-qubit Z feature map with an EfficientSU2 ansatz under the identical optimiser, so a gain can be credited to the circuit and not to Adam.

VAR.	WHAT CHANGES	PARAMS	CV MACRO-F1
A	Fixed encoding, $\langle Z_0\rangle$ readout only	26	0.650 ± 0.044
B	A, CNOTs reversed	26	0.560 ± 0.011
C	B + trainable input scale, shift	74	0.836 ± 0.034
D	C + readout on all four qubits	77	0.825 ± 0.014
D′	D, CNOTs as in A **(selected)**	77	0.852 ± 0.005
E	D, four layers	101	0.833 ± 0.049
G	D, five layers	125	0.721 ± 0.070
F	E, chain in place of star	101	0.701 ± 0.047
H	Z map + EfficientSU2, 8 qubits	34	0.559 ± 0.035

Table 2. Variational ablation: mean ± sample SD of validation macro-F1 over the three folds. Four qubits, three layers and producer→consumer CNOTs unless stated; H is the control.

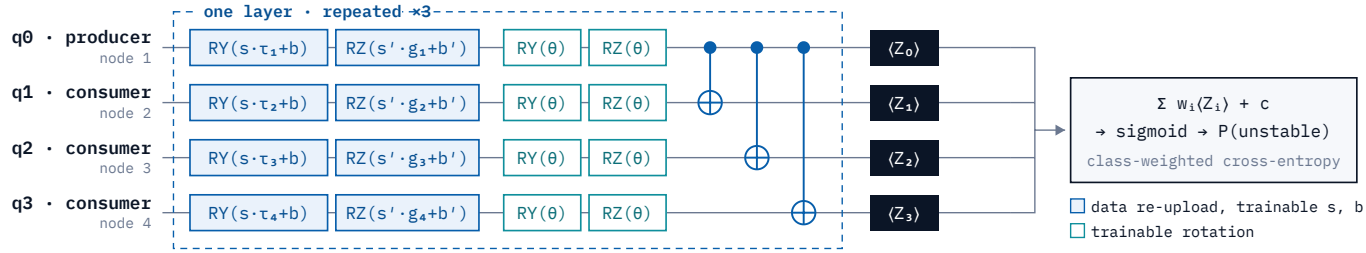


Figure 1. The selected variational classifier (variant D′): one qubit per grid node, three re-uploading layers, CNOTs along the grid's star.

04 Results

MODEL	QUBITS	CV MACRO-F1 (MEAN ± SD)	HOLDOUT MACRO-F1	HOLDOUT ACCURACY	HOLDOUT ROC-AUC
Classical RBF SVM (tuned C, γ)	–	0.880 ± 0.024	**0.902**	0.91	0.972
Gradient boosting (next-best classical)	–	0.829 ± 0.008	0.866	0.88	0.962
Angle-kernel SVM (selected encoding)	8	0.854 ± 0.037	**0.813**	0.84	0.934
Dense-angle-kernel SVM	4	0.825 ± 0.062	*not selected, so never run on the holdout*		
Amplitude-kernel SVM	3	0.571 ± 0.034	*not selected, so never run on the holdout*		
Variational circuit D′ (selected variant)	4	0.852 ± 0.005	**0.830**	0.84	0.916
EfficientSU2 control H	8	0.559 ± 0.035	*not selected, so never run on the holdout*		
Majority-class dummy	–	0.389 ± 0.001	0.390	0.64	0.500

Table 3. Same features, rows and folds for every model. SD: sample SD over three folds. Quantum models: noiseless statevector simulation.

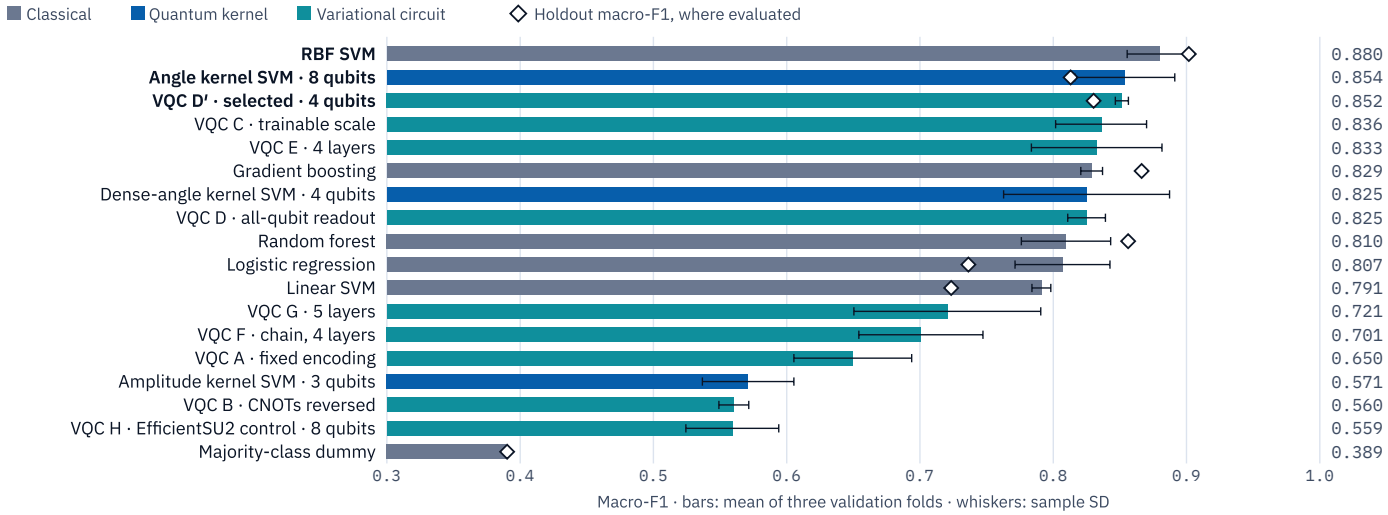


Figure 2. Every model we trained, ranked by cross-validation macro-F1. Diamonds: the one-time holdout score, where a model was run on it.

The classical RBF SVM leads on every holdout metric. The two quantum finalists tie on accuracy (84 of 100 rows) but behave differently: the angle kernel catches 95% of unstable grids and only 64% of stable ones, while the variational circuit, trained with class weights, is balanced (83% and 84%). Dense-angle encoding gives up little for half the qubits (0.825 against 0.854 in cross-validation). Amplitude encoding collapses to 0.571.

How strong is the evidence?

All three finalists were scored on the same 100 rows, so we compare them row by row (Table 4). The exact McNemar test looks only at rows where one model is right and the other wrong; the bootstrap resamples the 100 rows 10,000 times.

A VS B	A ONLY	B ONLY	P	Δ F1
Classical vs angle kernel	8	1	0.039	+0.089
Classical vs variational	13	6	0.167	+0.072
Variational vs angle kernel	10	10	1.000	+0.017

Table 4. Paired holdout tests: rows only one model gets right, exact McNemar p and macro-F1 difference. 95% bootstrap intervals for the differences: +0.026 to +0.164; −0.019 to +0.165; −0.083 to +0.117.

The classical lead over the angle kernel is supported ($p = 0.039$), though it would not survive a correction for three comparisons. The lead over the variational circuit is not statistically reliable at this sample size ($p = 0.167$), and the two quantum models cannot be told apart. A 100-row holdout ranks models only roughly, and we report it as such.

Context. Trained on all 8,000 rows, an RBF SVM reaches macro-F1 0.985 on 2,000 test rows. That is not a matched comparison, but it shows how far the 400-row limit, which exists only because circuits are slow to simulate, holds every model back.

05 What we learned

1 · Our kernel encodings are classical formulas

None of the three encodings applies a gate between two data qubits. Angle and dense-angle states are therefore product states, and the amplitude state is just the normalised input vector. Their fidelity kernels have closed forms:

$$K_{angle}(x, z) = \prod_i \cos^2((x_i - z_i)/2)$$
$$K_{amp}(x, z) = (x \cdot z)^2$$

We checked both, and the dense-angle analogue, against the simulator: they agree to 2.4×10^{-13}. These are valid quantum models, and running them on hardware would be a real experiment. In simulation, though, a quantum-kernel SVM built this way is a classical kernel SVM drawn as a circuit. It cannot show a quantum effect. The angle kernel is a periodic relative of the RBF kernel with a fixed width, consistent with it landing close to, and just below, the tuned RBF SVM.

2 · Amplitude encoding loses the sign of the data

$(x \cdot z)^2$ cannot tell x from −x, and standardised features are centred on zero, so opposite operating points become identical. On the same unit-norm inputs, in cross-validation, the amplitude kernel scores 0.584, a plain linear kernel $x \cdot z$ scores 0.783 and an RBF kernel 0.814. Normalising the rows costs little; squaring the overlap costs about 0.20. The saving of five qubits was paid for with information the classifier needed.

3 · Entangling gates did not help the kernels

The encoding module also has three feature maps with RZZ entangling gates and re-uploading, whose kernels do not factorise: `grid_coupled` ties each node's reaction time to its own price coefficient, `custom_entangled` entangles a ring, and `ultra_dense` packs three features per qubit. Their designer proposed a bandwidth grid $\alpha \in \{0.1, 0.25, 0.5, 1\}$ that had never been run; we ran it on the training folds (Figure 3). Tuning helps: `grid_coupled` rises from 0.783 at the default $\alpha = 0.5$ to 0.839 at $\alpha = 0.1$, `custom_entangled` peaks at 0.793 and `ultra_dense` at 0.668. But all three do best at the smallest scales, where the entangling term has faded, and the best still sits below plain angle encoding (0.864, or 0.871 at half scale). Repetitions stayed at the default of two, so this is a result about these maps, not about entanglement in general.

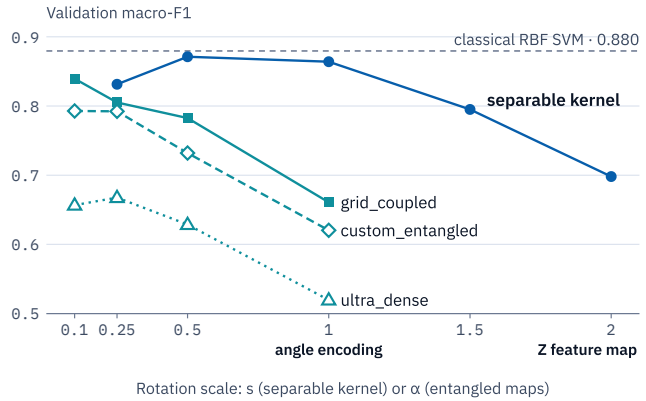


Figure 3. Rotation-scale sweep on the three training folds, at the better of C = 1 and C = 10 for each point. s scales the rotations of the separable kernel, α those of the entangled maps. The holdout was not used.

4 · The rotation scale is the knob that matters

How far a feature rotates a qubit decided more than any circuit choice. For the separable kernel $\prod_i \cos^2(s(x_i - z_i)/2)$, cross-validation gives 0.831, 0.871, 0.864, 0.795 and 0.698 at s = 0.25, 0.5, 1, 1.5 and 2 (Figure 3). Angle encoding is s = 1. The Z feature map of the team's first kernel experiment (a Hadamard, then a phase of 2x) is the same kernel at s = 2: rotating twice as far, and nothing else, takes it from 0.864 to 0.698.

The variational circuit shows the same thing from the other side. With fixed encoding angles it scored 0.650, or 0.560 with reversed CNOTs, no better than the EfficientSU2 control (0.559). Letting it learn a scale and shift for every uploaded feature lifted that circuit from 0.560 to 0.836, the largest effect in the study, in line with the view that encoding gates fix which frequencies a circuit can express[5]. Reading all four qubits with producer-to-consumer CNOTs gave the selected 0.852. More depth did not pay: four and five layers scored 0.833 and 0.721 against 0.825 with three, while the training loss kept falling (0.087, 0.050, 0.031), and a chain in place of the star scored 0.701.

5 · A matched tuning budget narrows the kernel gap

The classical SVM was tuned over C and γ; our kernel SVMs used C = 1 and a fixed scale. With the same two values of C the angle kernel moves from 0.854 to 0.864. With the scale tuned as well, the angle-family kernel reaches 0.871, against 0.880 for the classical RBF SVM on identical inputs.

> Every number in points 2 to 5 is training-fold cross-validation. The kernel sweeps are exploration added after the main protocol; we did not rerun the holdout for them, so the holdout scores in Table 3 stand as planned.

06 So, is quantum worth it here?

Not on this problem, in simulation. A tuned classical SVM is more accurate, trains in a fraction of a second and needs no simulator. The quantum models are close at 400 training rows, and the variational circuit gets there with four qubits, but close on a small sample is not an advantage. What we can say is where the quantum design choices mattered: the encoding scale mattered most, the qubit count hardly at all (four qubits matched eight), and entanglement and depth not at all or for the worse.

What would change the answer. Three things we did not do: tune the repetition count or entangling structure of the entangled maps, repeat the comparison over several splits and seeds, and add shot noise or a hardware run. Quantum would start to be worth it here only if an entangled kernel, with no simple classical formula, beat the classical SVM under that stricter protocol.

Limits. One simulated dataset and one grid topology; one split and seed; three folds; a 100-row holdout; nine variational variants compared on three folds, so the selected variant's cross-validation score is optimistic; p-values uncorrected for three comparisons; noiseless statevector simulation with no shot noise and no hardware run. We claim no quantum advantage.

07 How we used AI tools

OpenAI Codex assisted with the data pipeline, the classical benchmarks, the kernel and variational notebooks, debugging and documentation. Anthropic Claude generated the initial variational circuit design, which the ablation then tested. Anthropic Claude Code assisted the coordinator with repository integration, the reproducibility check, the results-analysis notebook (paired tests, closed-form kernel checks, exploration) and drafting this report, the slides and the project page. Every reported number is computed by the project code from the dataset; no AI tool produced or altered a score. Team members reviewed the work.

08 Who did what

Ziad Ahmed	Data engineering, leakage controls, classical baselines.
Seif Haytham	Quantum encodings, state-preparation validation, circuit resource analysis.
AbdUlRahman Hany	Quantum kernels, variational classifier and ablation.
Abdelrahman Gabr	Coordination, literature and design rationale, integration and reproducibility, results analysis, report, presentation.

09 References

[1] V. Arzamasov, *Electrical Grid Stability Simulated Data*, UCI Machine Learning Repository (2018). doi:10.24432/C5PG66. CC BY 4.0.

[2] B. Schäfer et al., "Taming instabilities in power grid networks by decentralized control", *Eur. Phys. J. Spec. Top.* 225 (2016).

[3] V. Havlíček et al., "Supervised learning with quantum-enhanced feature spaces", *Nature* 567 (2019).

[4] A. Pérez-Salinas et al., "Data re-uploading for a universal quantum classifier", *Quantum* 4, 226 (2020).

[5] M. Schuld, R. Sweke, J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models", *Phys. Rev. A* 103, 032430 (2021).

[6] scikit-learn developers, "Common pitfalls and recommended practices" and "Cross-validation", scikit-learn 1.8 documentation.

[7] IBM Quantum Learning, *Introduction to Quantum Machine Learning*. Software: Qiskit, PennyLane, scikit-learn.