Journal of Next Generation Technology (JNxtGenTech)
Vol. 6, Issue 4, August 2026

Quantum-Kernel Support Vector Machines for Zero-Day Intrusion Detection under Severe Class Imbalance in Industrial IoT Traffic

A controlled simulation study on a labelled synthetic benchmark

Varun Kaza[1]*, Fiona Shabani[2], Artur Make[2]

[1]Department of Computer Science and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, India.
[2]Department of Mechanics, Polytechnic University of Tirana, Albania;
Email idL: varunkaza20@gmail.com[1*], fionashabani11@gmail.com[2], artur.make@yahoo.com[3]

Article Received: 25 June 2025 *Article Accepted: 20 July 2026* *Article Published: 01 Aug 2026*

Citation
Varun Kaza, Fiona Shabani, Artur Make, "Quantum-kernel SVMs for zero-day intrusion detection in industrial IoT traffic", Journal of Next Generation Technology (ISSN: 2583-021X), 6(4), pp. 1-16, August 2026.

Abstract

Industrial IoT networks make learned intrusion detection hard in two ways at once: attacks are rare, and the attacks that matter most are those absent from any training set. Quantum kernels have been proposed as a source of richer similarity functions from few qubits, yet they are seldom examined under these two conditions in a setting where the ground truth is known. We embed exact-statevector fidelity kernels in class-weighted and one-class support vector machines and evaluate them on a synthetic flow generator with three benign device profiles and five attack families, so that imbalance, attack novelty and hardware noise can be varied independently. Over ten seeds, a six-qubit ZZ-map SVM is statistically indistinguishable from a six-component RBF-SVM when attacks are rare (F1 0.711 against 0.593 at 100:1, $p = 0.32$; the classical model collapsed to flagging everything in two of ten runs), but it stays below classical models that see all twelve features. In leave-one-family-out tests, a benign-only quantum-kernel one-class SVM reached a mean AUROC of 0.901 and recall of 0.763 at a 5% false-positive rate, ahead of a one-class RBF with the library-default bandwidth by 0.055 and 0.079. That margin disappears once the RBF bandwidth is widened (best 0.924 and 0.789), so we attribute it to bandwidth rather than to a quantum effect. No method detected the stealthiest replay family. Kernel variance of the ZZ map falls exponentially with qubit count, and finite-shot sampling, not global depolarisation, is the dominant hardware risk. All results come from simulation on synthetic data and should be read as a methodological baseline, not as evidence of deployed performance.

Keywords: quantum kernel, support vector machine, intrusion detection, zero-day attacks, class imbalance, industrial IoT

I. Introduction

Programmable logic controllers, telemetry gateways and operator consoles now share networks that were once physically isolated. The traffic they produce is regular and dominated by a few device behaviours, which makes anomaly detection attractive, but it also means that labelled attacks are scarce compared with normal operation, often by a wide margin. Public intrusion data sets for enterprise and IoT settings [1–3] are a valuable starting point, yet an evaluation on any fixed collection says little about the situation a defender actually faces: a new attack family appears that no training example resembles. Machine-learning detectors that look excellent on random splits are widely cautioned to degrade when that closed-world assumption breaks [4].

Kernel methods are a natural fit for small, imbalanced problems, because a support vector machine needs only a similarity function between flows [5]. A quantum computer offers an unusual family of similarity functions: encode each flow as a quantum state and take the squared overlap of two states

as the kernel [6, 7]. Such kernels are, in principle, hard to evaluate classically for some circuit families [6, 8], although hardness alone does not imply usefulness on real data [9, 10]. Two obstacles are well documented: kernel values can concentrate around a constant as qubits are added, which removes the signal an SVM needs [11], and every kernel entry on hardware is a noisy sample rather than an exact number [12, 13].

This paper asks a narrow, practical question. *When attacks are rare and new, does a small quantum-kernel SVM offer anything that a well-tuned classical kernel does not?* We answer it with simulation, so that every factor can be controlled and every claim can be checked against a known ground truth. The pipeline is summarised in Fig. 1. Our contributions are:

- A reproducible synthetic generator of industrial flow records with three benign device profiles and five attack families, including a tunable share of camouflaged ("stealth") attack flows.
- A controlled comparison of a six-qubit ZZ-map quantum-kernel SVM with linear, RBF and random-forest baselines at benign-to-attack ratios from 1:1 to 100:1, with paired non-parametric tests over ten seeds.
- A leave-one-attack-family-out protocol for zero-day detection, scored by AUROC and by recall at a false-positive rate calibrated on held-out benign data, including benign-only one-class variants.
- A bandwidth-sensitivity analysis showing that an apparent quantum advantage in one-class detection is reproduced by a classical RBF kernel with a wider bandwidth, together with measurements of kernel concentration, shot noise and depolarisation.

The negative and mixed findings are reported with the same prominence as the positive ones, since for a method with so many degrees of freedom the honest map of where it does not help is as useful as a headline number.

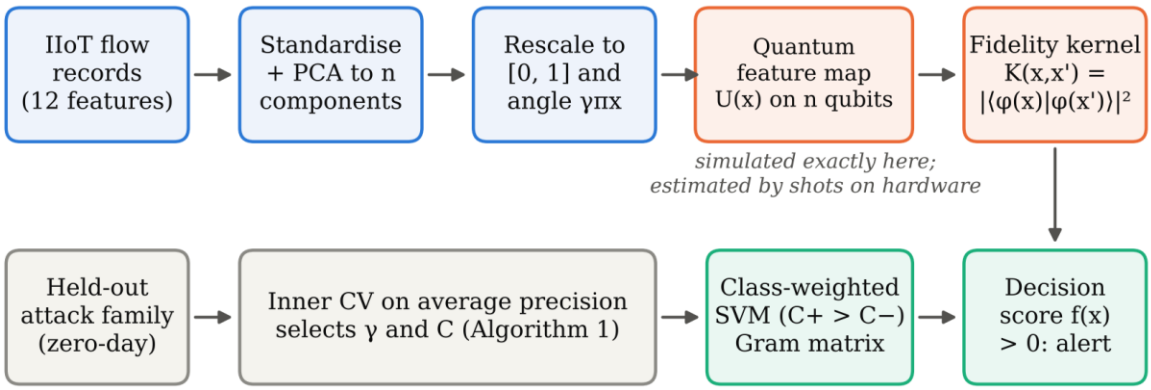

Fig. 1. Overview of the proposed pipeline. Flow features are standardised, projected to n principal components and rescaled to [0,1]; a feature-map circuit prepares a state per flow; squared overlaps form the Gram matrix used by a class-weighted SVM (supervised branch) or a one-class SVM (benign-only branch), and alerts are raised from calibrated scores.

II. Background and Related Work

Quantum kernels. Feature-map kernels were introduced for supervised learning in [6] and placed in the general framework of feature Hilbert spaces in [7]. The inductive bias of such kernels, and the role of the data distribution in deciding whether they can beat classical ones, has been analysed in [10] and [9]. Exponential concentration of kernel values, driven by expressivity, entanglement, global

measurements or noise, was characterised in [11]; it parallels the barren-plateau phenomenon in variational circuits [14, 15]. Data re-uploading [16] and the choice of encoding [17] determine which functions a circuit can represent, and trainable embedding kernels have been demonstrated on near-term devices [18]. Reviews of the field and its open problems are given in [13, 19].

Intrusion detection under imbalance and novelty. Anomaly detection, including one-class formulations that model only normal behaviour, is surveyed in [20]; the one-class support vector machine of [21] is the classic kernel instance. The practical difficulty of deploying learned detectors outside the data they were trained on is discussed in [4]. For evaluation, precision–recall analysis is preferred to ROC analysis when positives are rare [22], and the Matthews correlation coefficient is more informative than accuracy or F1 for imbalanced binary outcomes [23]. We follow these recommendations. The literature that combines quantum kernels with intrusion detection specifically is young and heterogeneous, and we do not attempt a survey of it; instead we build on the established foundations above and concentrate on a clean, controlled comparison.

III. Quantum-Kernel Classification: Formulation

A. Fidelity kernels

Let $\mathbf{x} \in \mathbb{R}^{12}$ be a flow record. After preprocessing (Section V-B) it becomes $\tilde{\mathbf{x}} \in [0,1]^n$, where n is the number of qubits. A parameterised circuit $U(\mathbf{x})$ prepares

$$|\phi(\mathbf{x})\rangle = U(\mathbf{x})\,|0^{\otimes n}\rangle, \tag{1}$$

and the fidelity kernel is the squared overlap of two such states [6, 7],

$$k(\mathbf{x}, \mathbf{x}') = |\langle \phi(\mathbf{x}) \mid \phi(\mathbf{x}')\rangle|^2 = \left|\langle 0^{\otimes n} \mid U^{\dagger}(\mathbf{x})\,U(\mathbf{x}') \mid 0^{\otimes n}\rangle\right|^2. \tag{2}$$

On hardware, $k(\mathbf{x}, \mathbf{x}')$ is the probability of reading the all-zero string after running $U(\mathbf{x}')$ followed by $U^{\dagger}(\mathbf{x})$. Because it is a squared inner product in the Hilbert space of the circuit, the resulting Gram matrix is symmetric, positive semi-definite and has unit diagonal.

B. Feature maps

Features enter the circuit as rotation angles $\varphi_i = \gamma\pi\,\tilde{x}_i$, where the bandwidth γ is a hyper-parameter:

$$\varphi_i = \gamma\,\pi\,\tilde{x}_i, \qquad \tilde{x}_i \in [0,1]. \tag{3}$$

The primary map is of the IQP type [6, 24]: L repetitions of a Hadamard layer followed by a diagonal phase unitary that contains single-qubit terms and products of neighbouring features on a ring E of qubit pairs,

$$U_{\mathrm{ZZ}}(\mathbf{x}) = \left[\,D(\mathbf{x})\,H^{\otimes n}\right]^{L}, \tag{4}$$

$$D(\mathbf{x}) = \exp\!\left(i\left[\textstyle\sum_{i=1}^{n}\varphi_i\,Z_i + \sum_{(i,j)\in E}(\pi-\varphi_i)\,(\pi-\varphi_j)\,Z_iZ_j\right]\right). \tag{5}$$

As a contrasting design we use a re-uploading hardware-efficient (HEA) map [16, 17], in which each layer applies R_Y and R_Z rotations driven by neighbouring features and a ring of controlled-Z gates,

$$V(\mathbf{x}) = \mathrm{CZ}_{\mathrm{ring}}\textstyle\prod_{i=1}^{n} R_Z\,(\varphi_{i+1})\,R_Y(\varphi_i), \qquad U_{\mathrm{HEA}}(\mathbf{x}) = V(\mathbf{x})^{L}, \tag{6}$$

with indices taken modulo n. Fig. 2 shows both circuits for $n = 4$. The ZZ map generates feature products, so its kernel is not a function of the distance between two flows alone; the HEA map is closer to a smooth, locally varying kernel.

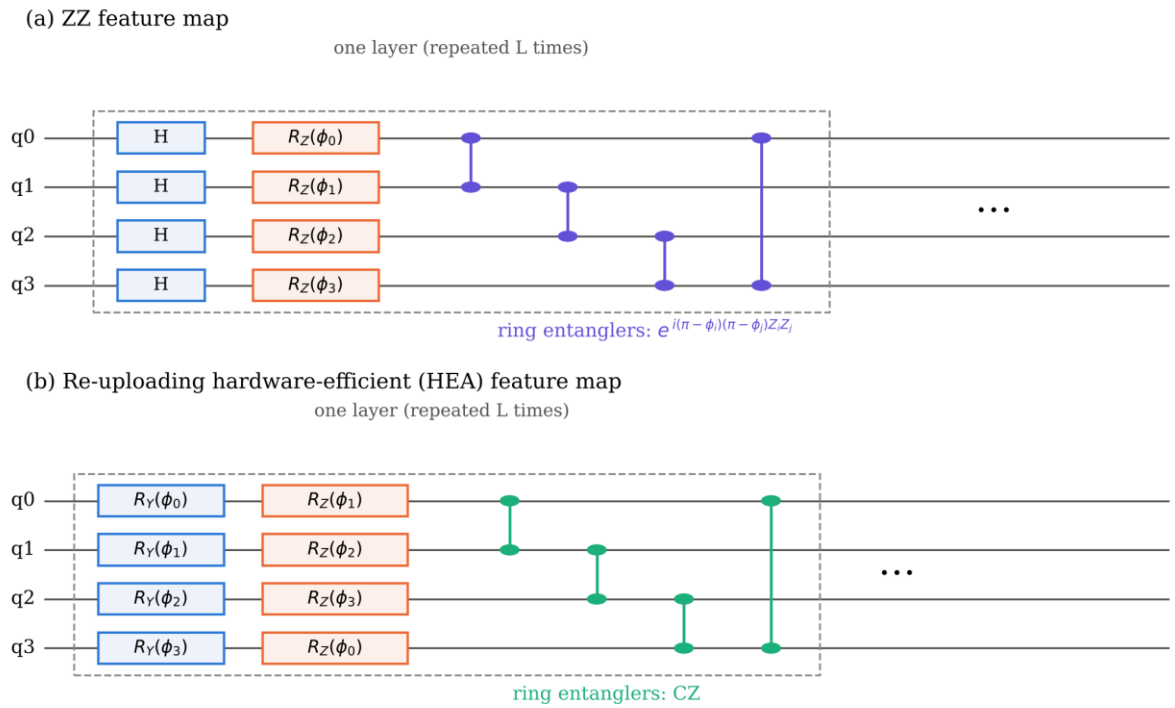

Fig. 2. Circuit diagrams of the two feature maps for $n = 4$ qubits and one repetition. (a) ZZ map: Hadamard layer, single-qubit phases and neighbouring-pair phase gates on a ring. (b) Re-uploading HEA map: R_Y/R_Z rotations followed by a ring of CZ gates. Dashed boxes mark the block repeated L times.

C. Class-weighted and one-class SVMs on precomputed Gram matrices

Given the Gram matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ of N training flows with labels $y_i \in \{-1,1\}$ (attack: $y_i = 1$), the soft-margin SVM solves the dual problem [5]

$$\max_{\boldsymbol{\alpha}} \ \sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij} \quad \text{s.t.} \quad 0 \le \alpha_i \le C_{y_i}, \ \ \sum_i \alpha_i y_i = 0. \tag{7}$$

To counter imbalance the box constraint is scaled per class so that both classes carry equal total weight,

$$C_{\mathrm{a}} = \frac{N}{2N_{\mathrm{a}}} C, \qquad C_{\mathrm{b}} = \frac{N}{2N_{\mathrm{b}}} C, \tag{8}$$

where N_{a} and N_{b} are the numbers of attack and benign training flows, and C_{y_i} in Eq. (7) equals C_{a} for attacks and C_{b} for benign flows. The decision score is $f(\mathbf{x}) = \sum_i \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b$, and a flow is flagged when $f > 0$. For benign-only training we use the one-class SVM of [21] with a precomputed kernel and parameter ν (an upper bound on the fraction of training points treated as outliers); the anomaly score is the negated decision function.

D. Kernel concentration, alignment and noise

Concentration is measured by the variance of off-diagonal Gram entries on a sample of flows. If the variance falls as 2^{sn} with $s < 0$, the Gram matrix approaches a constant matrix and generalisation is lost [11]. We estimate s from the slope of a linear fit,

$$\log_2 \sigma_n^2 = a + s\, n, \qquad \sigma_n^2 = \mathrm{Var}_{i \ne j}[k(\mathbf{x}_i, \mathbf{x}_j)]. \tag{9}$$

Kernel–target alignment [25] compares the centred Gram matrix K_c with the centred ideal kernel T_c, $T = \mathbf{t}\mathbf{t}^{\top}$, $t_i = \pm 1$:

$$\mathrm{KTA}(K, \mathbf{y}) = \frac{\langle K_c, T_c \rangle_F}{\|K_c\|_F \, \|T_c\|_F}. \tag{10}$$

Two hardware effects are modelled. With S shots per entry, the estimate is binomial, and a global depolarising channel of fidelity λ mixes the kernel with its maximally mixed value,

$$\hat{k}_{ij} \sim 1/S \ \text{Binomial}(S,\, k_{\lambda,ij}), \qquad k_\lambda = \lambda\, k + (1-\lambda)\, 2^{-n}. \tag{11}$$

The sampled Gram matrix need not be positive semi-definite, so it is projected onto the PSD cone by clipping negative eigenvalues,

$$\widetilde{K} = V \max(\Lambda, 0)\, V^{\top}, \qquad \widehat{K} = V \Lambda V^{\top}. \tag{12}$$

An analytic observation is useful later. If a kernel is replaced by $\lambda k + c$, the quadratic term of the dual objective changes to

$$\textstyle\sum_{i,j} \alpha_i\, \alpha_j y_i y_j \big(\lambda K_{ij} + c\big) = \lambda \sum_{i,j} \alpha_i\, \alpha_j y_i y_j K_{ij} + c\big(\sum_i \alpha_i\, y_i\big)^2 = \lambda \sum_{i,j} \alpha_i\, \alpha_j y_i y_j K_{ij}, \tag{13}$$

because $\sum_i \alpha_i\, y_i = 0$. A global depolarising channel therefore only rescales C and leaves the SVM solution unchanged once C is re-selected; it cannot by itself explain a loss of accuracy.

IV. Proposed Framework

Algorithm 1 describes the supervised branch. All preprocessing statistics are estimated on training data only. The bandwidth γ and the regularisation C are chosen jointly by stratified inner cross-validation, scored by average precision, so that the selection criterion matches the imbalanced setting rather than accuracy. The number of folds is capped at three so that the rarest class still appears in every fold.

Algorithm 1. Training of the quantum-kernel SVM (supervised branch).

Input: labelled flows $(X, \mathbf{y})$; qubits n; depth L; bandwidth grid Γ; regularisation grid $\mathcal{C}$; folds $F = 3$.
Output: preprocessing transform T, selected $(\hat{\gamma}, \hat{C})$, trained SVM.

Fit standardiser, PCA(n) and min–max scaler on X; obtain $\widetilde{X} = T(X)$ clipped to [0,1].
for each $\gamma \in \Gamma$ **do**
    Compute states $U(\gamma \pi \tilde{\mathbf{x}}_i)\,|0^{\otimes n}\rangle$ for all training flows (Eqs. (1), (4)–(5)).
    Form $K^{\gamma}_{ij} = \left|\langle \phi_i \mid \phi_j \rangle\right|^2$.
for each $(\gamma, C) \in \Gamma \times \mathcal{C}$ **do**
    Split into F stratified folds; fit a class-weighted SVM on K^{γ} of the training folds (Eq. (8)).
    Score the held-out fold by average precision; average over folds.
$(\hat{\gamma}, \hat{C}) \leftarrow$ pair with the highest mean average precision.
Fit the final class-weighted SVM on $K^{\hat{\gamma}}$ with $\hat{C}$; **return** $T, \hat{\gamma}, \hat{C}$, SVM.

Algorithm 2 describes the benign-only branch used for zero-day detection. Its alert threshold is not the library default but a quantile of anomaly scores on a separate benign validation set, so that the false-positive rate is fixed by construction.

Algorithm 2. Calibrated zero-day detection with a benign-only quantum-kernel one-class SVM.

Input: benign training flows X_b; benign validation flows X_v; n, L, γ, ν; target false-positive rate α.
Output: alert decision for each incoming flow.

1: Fit the preprocessing transform T on X_b only.
2: Compute states for $T(X_b)$ and the Gram matrix K_b (Eq. (2)).
3: Fit a one-class SVM with kernel K_b and parameter ν.
4: Score validation flows: $s(\mathbf{x}) = -f(\mathbf{x})$ for $\mathbf{x} \in X_v$; set τ_α to the $(1-\alpha)$ quantile of these scores.
5: **for** each incoming flow **x do**
6: Compute its overlaps with the stored training states; obtain $s(\mathbf{x})$.
7: Raise an alert if $s(\mathbf{x}) > \tau_\alpha$.

Cost. Simulating N states of n qubits costs $O(NLn2^n)$ operations and the Gram matrix $O(N^2 2^n)$. On hardware, the training Gram matrix needs $N(N-1)/2$ circuit pairs, each repeated S times:

$$N_{\text{shots}} = S\,\frac{N(N-1)}{2}. \tag{14}$$

For $N = 1200$ and $S = 512$ this is about 3.7×10^8 shots for training alone, and classifying a test flow costs a further NS shots. These figures motivate the shot-noise experiment in Section VI-F.

V. Experimental Setup

A. Synthetic industrial-flow generator

No proprietary or public capture is used. Flow records are drawn from a generator written for this study, so that every property of the data is known. Each record has twelve features (Table I). Benign traffic mixes three device profiles with prior probabilities 0.45 (PLC polling), 0.35 (MQTT telemetry) and 0.20 (HMI sessions), each with its own feature means and spreads. A shared latent factor with strength 0.35 correlates the noise across features so that the data are not axis-aligned.

Table I. Flow features produced by the generator.

#	Feature	Meaning
1	log_duration	Logarithm of flow duration
2	log_fwd_pkts	Logarithm of forward packet count
3	log_bwd_pkts	Logarithm of backward packet count
4	fwd_pkt_size	Mean forward packet size (bytes)
5	bwd_pkt_size	Mean backward packet size (bytes)
6	log_iat_mean	Logarithm of mean inter-arrival time
7	iat_cv	Coefficient of variation of inter-arrival time
8	syn_ratio	Fraction of packets carrying SYN
9	payload_entropy	Payload entropy (bits per byte)
10	log_dst_ports	Logarithm of distinct destination ports
11	func_code_diversity	Diversity of industrial function codes
12	write_ratio	Share of write operations

Five attack families are generated (Table II). A fraction of each family is "stealth" traffic: a benign record shifted by only 30% of the way towards the family mean, $\mathbf{x} = \mathbf{x}_b + 0.3(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)$. Stealth fractions rise from 0.05 for the flooding attack to 0.35 for exfiltration, so the families span a range of

difficulty. Fig. 3 projects a sample onto its first two principal components (41.3% and 22.0% of the variance).

Table II. Attack families in the generator and their stealth fractions.

Family	Characteristic signature	Stealth share
DoS	Very high packet rate, high SYN ratio, tiny inter-arrival times	0.05
Scan	Many destination ports, high SYN ratio, small packets	0.10
Replay	Near-copy of PLC traffic with abnormally regular timing and slightly higher entropy	0.25
Inject	Elevated write ratio and function-code diversity on otherwise normal flows	0.30
Exfil	Long flows, large forward payloads, high entropy, long inter-arrival times	0.35

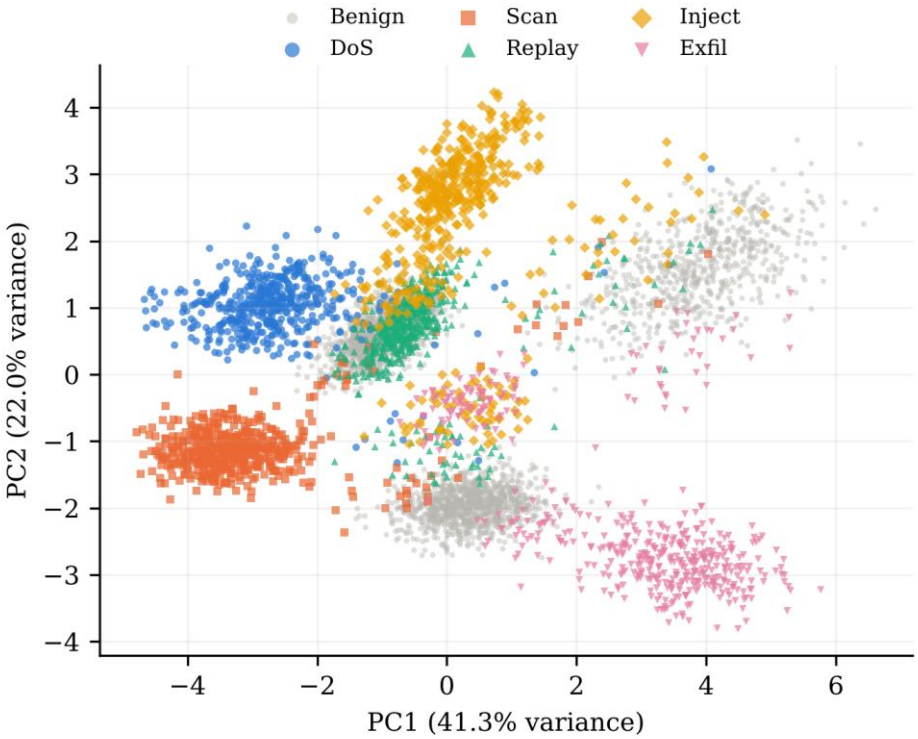

Fig. 3. Two-dimensional PCA projection of a synthetic sample. Benign traffic forms three device-profile clusters; attack families occupy distinct regions, but stealth samples of Replay, Inject and Exfil lie inside or at the edge of the benign mass.

B. Protocol

Training sets contain $N = 1200$ flows. For the imbalance study the benign-to-attack ratio $\rho \in \{1,5,10,25,50,100\}$ gives 600, 200, 109, 46, 24 and 12 attack flows, split evenly over the five families. Every test set holds 2400 benign and 240 attack flows (48 per family; prevalence 0.091), which is deliberately more favourable to precision than a 100:1 field ratio, so absolute precision should not be extrapolated. Features are standardised, reduced by PCA to $n = 6$ components and min–max scaled to [0,1] with clipping, all fitted on training data. Table III lists the models and settings. The primary quantum configuration (ZZ map, $n = 6$, $L = 2$) was fixed in advance; the ablation in Section VI-D shows it is not the best of the grid. Search grids were widened once, after a single calibration run on

an independent seed on which the selected values sat at the grid edge; no evaluation seed was used for tuning.

Table III. Models and settings. All SVMs use balanced class weights; selection is by three-fold stratified inner cross-validation on average precision.

Model	Input	Hyper-parameters searched / fixed
QK-SVM (ZZ)	6 PCA comps, 6 qubits	$\gamma \in \{0.1,0.25,0.5,1,2\}$, $C \in \{0.01, \ldots, 100\}$, $L = 2$, ring
RBF-SVM (6D)	6 PCA comps	kernel scale $\{0.1,0.3,1,3,10\}/d$, same C grid
RBF-SVM (12D)	12 standardised features	same grids as above
Linear-SVM (6D)	6 PCA comps	same C grid
Random forest (12D)	12 raw features	200 trees, min. leaf 2, balanced subsample [26]
OC-SVM RBF	6 PCA comps, benign only	$\nu = 0.05$, kernel scale $1/d$ (fixed)
OC-SVM QK	6 PCA comps, benign only	$\nu = 0.05$, $\gamma = 0.5$ (fixed), ZZ, $L = 2$

Zero-day protocol. For each of the five families in turn, that family is removed from training (training ratio 20:1 over the remaining four families) and appears only in the test set with 2400 benign flows. Scores are summarised by AUROC and by recall at a false-positive rate of 5%, with the threshold set as in Eq. (15):

$$\tau_\alpha = Q_{1-\alpha}\{s(\mathbf{x}): \mathbf{x} \in \mathcal{V}\}, \qquad R_\alpha = \frac{1}{|\mathcal{A}|}\sum_{\mathbf{x}\in\mathcal{A}} \mathbf{1}\,[s(\mathbf{x}) > \tau_\alpha], \quad \alpha = 0.05, \tag{15}$$

where $\mathcal{V}$ is a separate set of 1000 benign validation flows and $\mathcal{A}$ the unseen-family attacks. Calibrating on benign data is the only option a defender has for a family that has not been seen. Imbalanced results are further reported with the F1 score, the Matthews correlation coefficient [23] and average precision,

$$\mathrm{MCC} = \frac{\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}, \qquad \mathrm{AP} = \sum_k (R_k - R_{k-1})\, P_k, \tag{16}$$

with precision–recall analysis preferred over ROC analysis for rare positives [22]. All main experiments use ten seeds (the ablation five, the noise sweeps ten for exact and sampled kernels at $\lambda = 1$ and five for $\lambda < 1$, the concentration study three). Differences between two methods are tested with the paired Wilcoxon signed-rank test [27] over seeds, as recommended for comparing classifiers [28]. With ten pairs the smallest attainable two-sided p is 0.002, and no correction for multiple comparisons is applied, so p-values are indicative rather than confirmatory. Quantum states are simulated exactly with NumPy [29]; classical models use scikit-learn [30].

VI. Results and Discussion

A. Detection under increasing class imbalance

Fig. 4 and Table IV show the effect of imbalance on the supervised detectors. At balanced training, the quantum-kernel SVM scores F1 = 0.740, slightly below the six-component RBF-SVM (0.771; paired difference −0.031, $p = 0.002$) and well below the classical models with access to all twelve features (RBF 0.825, random forest 0.855). As attacks become rarer the picture changes in a specific way. At 50:1 the quantum model (0.727) is ahead of the same-input RBF (0.708) but the gap is within noise ($p = 0.38$); at 100:1 it is 0.711 against 0.593 ($p = 0.32$), and the large standard deviation of the classical baseline (±0.239) reveals the cause: with only 12 attack training flows, the six-dimensional RBF-SVM degenerated to flagging every flow in two of ten seeds (false-positive rate 1.0), whereas the quantum model never did. In these two runs the classical model failed outright; the quantum

kernel's selected bandwidth was almost always the smallest on the grid ($\gamma = 0.1$ in nine of ten seeds at 100:1), i.e. a smooth, heavily regularised kernel, which may be what keeps it stable.

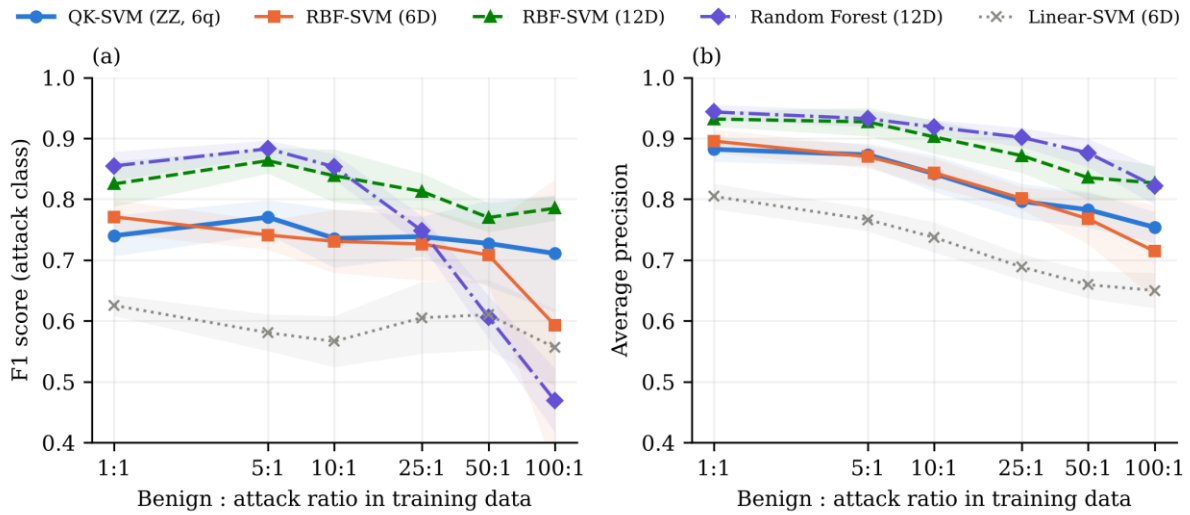

Fig. 4. Effect of training imbalance on (a) F1 of the attack class and (b) average precision; markers are means over ten seeds and shaded bands are ±1 standard deviation. Test prevalence is fixed at 0.091.

Table IV. F1 and average precision (AP) of the supervised detectors versus training imbalance ρ (benign : attack). Mean±SD over ten seeds.

Model	F1 1:1	AP 1:1	F1 10:1	AP 10:1	F1 50:1	AP 50:1	F1 100:1	AP 100:1
Linear (6D)	0.626±0.017	0.805±0.021	0.566±0.042	0.737±0.024	0.610±0.057	0.659±0.022	0.556±0.063	0.650±0.029
RBF (6D)	0.771±0.027	0.895±0.019	0.731±0.052	0.843±0.024	0.708±0.047	0.768±0.042	0.593±0.239	0.714±0.076
QK-SVM (6q)	**0.740±0.033**	**0.882±0.020**	**0.736±0.048**	**0.842±0.030**	**0.727±0.071**	**0.783±0.029**	**0.711±0.096**	**0.754±0.026**
RBF (12D)	0.825±0.037	0.932±0.013	0.838±0.043	0.902±0.027	0.770±0.024	0.835±0.036	0.785±0.020	0.827±0.029
Random forest	0.855±0.023	0.943±0.012	0.853±0.021	0.919±0.012	0.606±0.035	0.876±0.023	0.469±0.052	0.822±0.030

Models marked 6D/6q use six PCA components; 12D models use all features. Bold: the quantum-kernel model.

The classical models with full features are stronger throughout. At 100:1 they reach AP 0.827 (RBF) and 0.822 (random forest) against 0.754 for the quantum model (paired differences −0.073 and −0.068, $p = 0.002$), so six qubits fed by six principal components lose information that twelve raw features retain. The comparison that isolates the kernel is the one against the six-component RBF, and there the evidence is a modest, non-significant advantage at extreme imbalance and a small, significant disadvantage at balance. The random forest illustrates a different pitfall (Table V): at 100:1 its precision is 1.000 but its recall only 0.308, because a default decision threshold is poorly matched to the training prior, so F1 (0.469) overstates its inferiority while AP (0.822) shows that its ranking is good.

Table V. Detailed metrics at 100:1 training imbalance (12 attack flows). Mean±SD over ten seeds.

Model	Precision	Recall	FPR	MCC	AUROC
Linear (6D)	0.820±0.263	0.484±0.140	0.029±0.051	0.571±0.075	0.791±0.062
RBF (6D)	0.785±0.367	0.650±0.211	0.202±0.420	0.569±0.306	0.826±0.076
QK-SVM (6q)	**0.811±0.189**	**0.660±0.046**	**0.026±0.047**	**0.697±0.105**	**0.877±0.018**
RBF (12D)	0.968±0.059	0.663±0.037	0.003±0.005	0.785±0.022	0.921±0.021
Random forest	1.000±0.000	0.308±0.045	0.000±0.000	0.535±0.040	0.932±0.018

The precision–recall curves in Fig. 5 show the same ordering at a fixed 50:1 ratio: the two full-feature models dominate, and the quantum and six-component RBF curves are close, with the quantum curve slightly higher for recalls between 0.6 and 0.8.

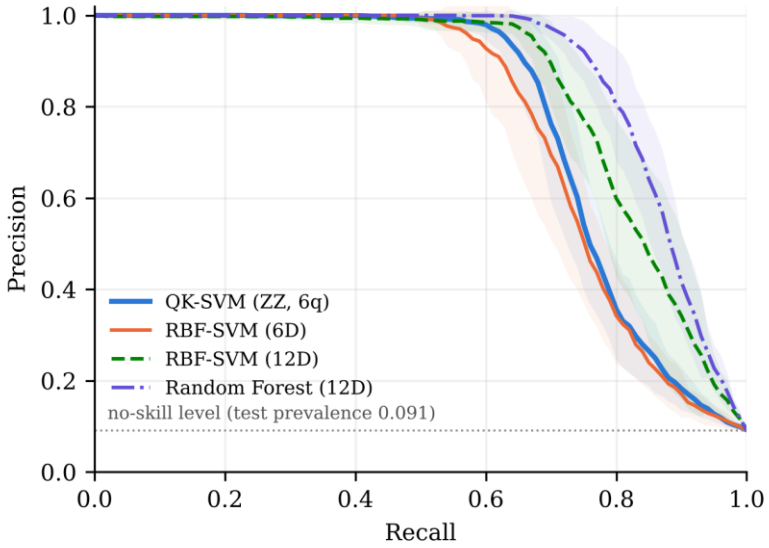

Fig. 5. Precision–recall curves at 50:1 training imbalance (mean over ten seeds, bands ±1 SD). The dotted line is the no-skill level.

B. Zero-day detection

Fig. 6 and Table VI report leave-one-family-out results. Three observations stand out. First, difficulty is family-specific: the scan family is found by every method (AUROC 0.97–1.00), whereas the replay family, whose flows resemble PLC traffic apart from timing regularity, is hardly separable from benign traffic by any method (AUROC 0.52–0.69, recall at 5% FPR below 0.21). Second, supervised models trained on the four remaining families often score a novel family below benign traffic: the six-component RBF-SVM obtains AUROC 0.35 on exfiltration, because its decision function extrapolates the "attack" label away from the regions it has seen. Third, models trained on benign traffic alone are the most robust.

Table VI. Zero-day detection (leave-one-family-out). Per-family columns give mean AUROC over ten seeds; the last two columns give AUROC and recall at 5% false-positive rate averaged over families (mean±SD over seeds).

Model	DoS	Scan	Replay	Inject	Exfil	Mean AUROC	Recall @5% FPR
Linear (6D)	0.55	0.99	0.52	0.47	0.28	0.563±0.086	0.267±0.062
RBF (6D)	0.67	0.98	0.54	0.68	0.35	0.642±0.107	0.392±0.116
QK-SVM (6q)	**0.76**	**0.97**	**0.61**	**0.86**	**0.64**	**0.768±0.067**	**0.482±0.092**
RBF (12D)	0.92	0.99	0.69	0.92	0.65	0.835±0.056	0.625±0.078
Random forest	1.00	1.00	0.68	0.85	0.50	0.804±0.033	0.518±0.043
OC-SVM RBF	0.99	1.00	0.59	0.89	0.77	0.846±0.011	0.684±0.017
OC-SVM QK	**0.99**	**0.99**	**0.61**	**0.96**	**0.96**	**0.901±0.010**	**0.763±0.011**

Supervised models are trained at 20:1 on the four remaining families; the two OC-SVM rows use only the benign training flows. Bold: quantum-kernel models.

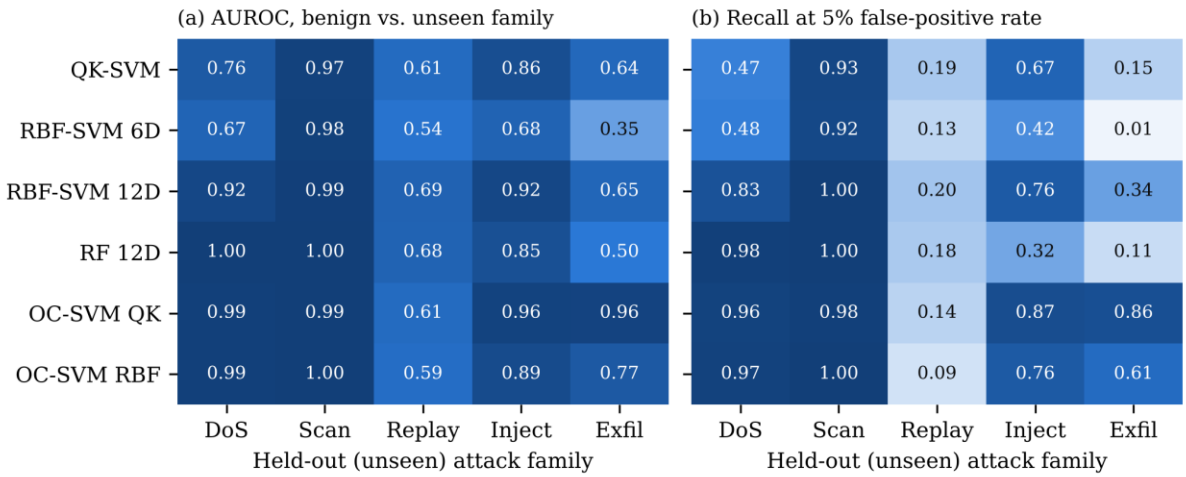

Fig. 6. Zero-day detection per held-out attack family. (a) AUROC and (b) recall at 5% false-positive rate, averaged over ten seeds. Darker cells indicate higher values.

Supervised quantum kernel. The quantum-kernel SVM improves on the six-component RBF-SVM by 0.126 in mean AUROC ($p = 0.02$; better in 8 of 10 seeds) and by 0.090 in recall at 5% FPR ($p = 0.11$, not significant), mainly by avoiding the inverted behaviour on exfiltration and by doing better on injection (0.86 against 0.68). Against the full-feature RBF-SVM it is lower by 0.066 in AUROC ($p = 0.19$) and by 0.143 in recall ($p = 0.01$). In short, with the same six inputs the quantum kernel generalises to unseen families somewhat better than a classical kernel, but it does not make up for the discarded features.

One-class models. The benign-only quantum-kernel OC-SVM has the highest mean AUROC (0.901±0.010) and recall at 5% FPR (0.763±0.011), and exceeds the one-class RBF-SVM with the library-default kernel scale by 0.055 and 0.079 (both $p = 0.002$, ten of ten seeds), the full-feature RBF-SVM by 0.067 and 0.138. The margin remains when replay is excluded (0.063 and 0.085, $p = 0.002$). Used with its default threshold, however, the quantum one-class model produces a false-positive rate of 0.168 against 0.051 for the RBF counterpart, so its raw alerts would be unusable; the calibrated recall at 5% FPR is the fair measure, and calibration on benign data fixes the problem at no labelling cost.

C. Is the one-class advantage a quantum effect?

The one-class comparison used a fixed quantum bandwidth ($\gamma = 0.5$) and the library-default RBF scale ($1/d$); neither was tuned, but the two defaults are not equally well matched to the data. We therefore swept both bandwidths over the same ten seeds and five families (Table VII). The quantum kernel is best at $\gamma = 0.25$–0.5 (AUROC 0.900–0.901) and degrades quickly at larger γ, as expected from a kernel that concentrates. The RBF kernel improves steadily with a narrower kernel (larger scale) and reaches AUROC 0.924 and recall 0.789 at scale $300/d$. The best classical result therefore exceeds the best quantum one, and the apparent advantage of Section VI-B comes from the default RBF bandwidth being far too wide for this data. This analysis picks the RBF scale using the test families themselves, so 0.924 is an optimistic upper bound for a defender, who cannot tune on an unseen family; the quantum default, by contrast, happened to work without tuning. We read this as weak evidence of robustness to bandwidth choice, not as evidence of a quantum advantage.

Table VII. Bandwidth sensitivity of the one-class SVMs (mean over 50 seed–family runs). RBF: kernel scale in units of $1/d$ with $d = 6$; QK: bandwidth γ of Eq. (3).

RBF scale	AUROC	Recall @5%	QK γ	AUROC	Recall @5%
0.1	0.844	0.680	0.1	0.857	0.727
0.3	0.844	0.681	0.25	0.900	0.759
1	0.846	0.684	0.5	0.901	0.763
3	0.849	0.692	1	0.847	0.462
10	0.855	0.711	2	0.807	0.367
30	0.863	0.732			
100	0.903	0.766			
300	0.924	0.789			
1000	0.911	0.771			

The default scale ($1/d$, i.e. 1) and the fixed quantum bandwidth (0.5) used in Table VI are the ones in the original comparison.

D. Feature-map design: qubits, depth and map type

Fig. 7 reports the performance of the ZZ and HEA maps over $n \in \{4,6,8\}$ qubits and $L \in \{1,2,3\}$ layers (five seeds, training ratio 20:1, standard test set). F1 ranges from 0.652 to 0.765, with seed-level standard deviations of 0.02–0.06 in most cells and 0.161 for ZZ with $n = 6$, $L = 3$, where at least one seed performed poorly. There is no monotone trend in depth or qubit number, and the primary configuration (ZZ, $n = 6$, $L = 2$; F1 0.734, AP 0.786) is not the best: HEA with $n = 8$, $L = 3$ reaches F1 0.765 and AP 0.841. The differences are of the same size as the seed variation, so we do not claim a preferred design. Neither kernel-target alignment (0.16–0.39) nor off-diagonal variance predicts performance across the 90 runs (Spearman correlation with AP: -0.13, $p = 0.21$ and 0.05, $p = 0.61$), which cautions against using these quantities as cheap model-selection criteria in this regime.

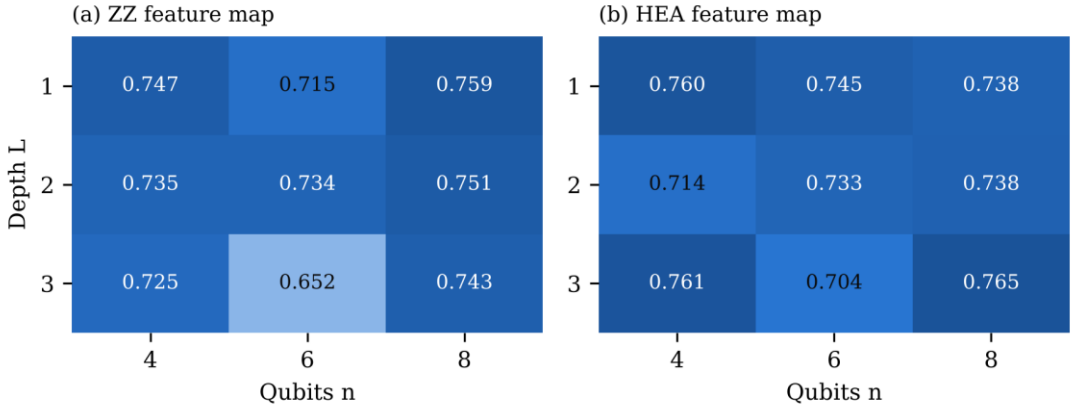

Fig. 7. Mean F1 of the quantum-kernel SVM for (a) the ZZ and (b) the HEA feature map, by qubit number and depth (five seeds). Cell values are printed; differences are small relative to seed variation.

E. Kernel concentration

Fig. 8 plots off-diagonal Gram variance against qubit count for $n = 2$ to 12. For the ZZ map the variance decays exponentially: the fitted exponent s in Eq. (9) is −0.47, −0.57 and −0.69 for $L = 1,2,3$ at $\gamma = 0.5$, and −0.75, −0.98 and −1.09 at $\gamma = 1$. A larger bandwidth and more layers concentrate the kernel faster, and at $\gamma = 1$, $L = 3$ the decay is about as steep as 2^{-n}. The HEA map is much flatter (s between −0.29 and +0.11). This is consistent with the analysis in [11] and explains why small

bandwidths were selected throughout: with $\gamma \leq 0.5$ and $n = 6$ the kernel is still informative, but scaling this particular map to many qubits without shrinking the bandwidth would flatten it. The measurement uses 400 flows and three seeds, so the exponents carry some uncertainty.

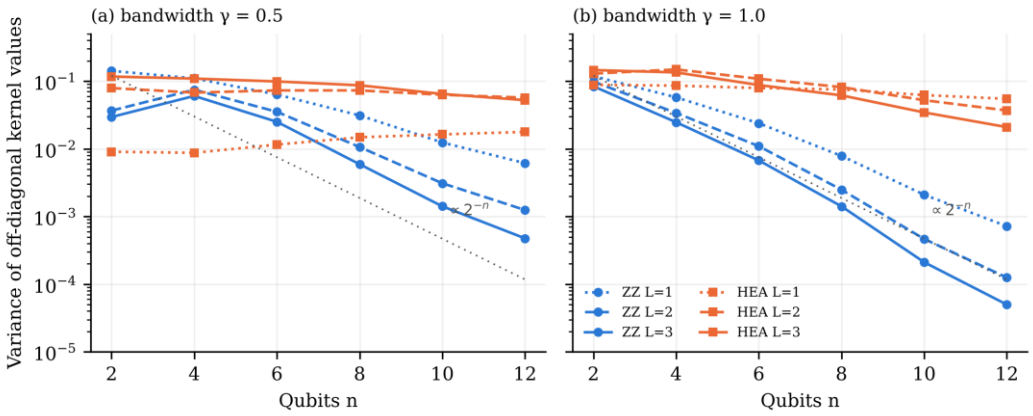

Fig. 8. Variance of off-diagonal kernel entries versus number of qubits (log scale) for ZZ (circles) and HEA (squares) maps at depths $L = 1,2,3$, for bandwidths (a) $\gamma = 0.5$ and (b) $\gamma = 1.0$. The dotted line shows 2^{-n} decay.

F. Finite shots and depolarising noise

Fig. 9 shows what happens when the exact kernel is replaced by hardware-like estimates (ZZ map, $n = 6$, 20:1 training ratio). With exact kernels the mean F1 is 0.767. Sampling with S between 32 and 8192 shots lowers it to between 0.628 and 0.727, with much larger spread in precision across seeds (standard deviation up to 0.20). The decline is not monotone in S: the mean at 512 shots (0.628) is lower than at 32 shots (0.688). With ten seeds the standard errors are about 0.02–0.03 for F1, so the ordering of nearby shot counts is not resolved, and we conclude only that shot noise at these budgets reduces and destabilises performance, not that a particular budget is optimal.

Global depolarisation is benign. At $\lambda = 0.4$ the exact-kernel F1 is 0.748, close to the 0.767 obtained at $\lambda = 1$ with a different seed set, in agreement with Eq. (13): the transformation $k \mapsto \lambda k + c$ leaves the solution unchanged apart from a rescaling of C. The damage in Fig. 9b therefore comes from the interaction of noise with finite sampling, where the signal $\lambda\,\Delta k$ shrinks against fixed binomial variance. For a fixed shot budget, hardware with higher fidelity helps because it enlarges the signal-to-noise ratio of each kernel entry, not because the SVM is sensitive to the bias.

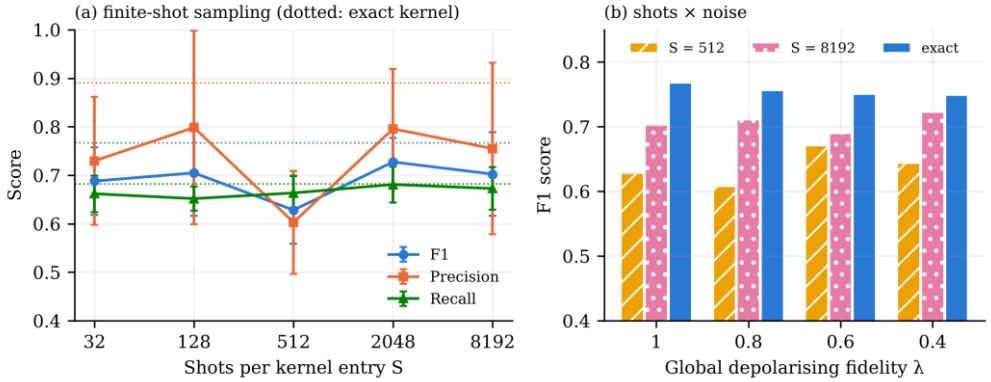

Fig. 9. (a) Quantum-kernel SVM performance versus shots per kernel entry (error bars ±1 SD over ten seeds; dotted lines: exact kernel). (b) F1 for combinations of depolarising fidelity λ and shots (five seeds for $\lambda < 1$).

G. Simulation cost

Table VIII. Wall-clock time of the exact simulation for $N = 1200$ flows (ZZ map, $L = 2$) on two CPU cores.

Qubits n	State preparation (s)	Gram matrix (s)	Total (s)
4	0.005	0.024	0.030
6	0.019	0.027	0.046
8	0.097	0.074	0.171
10	0.415	0.202	0.617
12	2.196	0.669	2.865

Table VIII confirms the expected growth of simulation cost by a constant factor per qubit, with about three seconds in total at twelve qubits. Exact simulation is cheap at the scales studied here; on hardware the cost lies instead in the shot budget of Eq. (14).

VII. Discussion and Limitations

What the evidence supports. (i) With six features, a quantum-kernel SVM is a reasonable, somewhat more stable alternative to a six-component RBF-SVM at extreme imbalance and for unseen families, although most differences are not statistically resolved with ten seeds. (ii) Neither six-component model reaches the accuracy of classical models that use all twelve features. (iii) Calibration on benign data matters more than the choice of kernel for zero-day alerting. (iv) The only large quantum-kernel advantage we observed, in one-class detection, is a bandwidth effect. (v) Kernel concentration and finite-shot noise, not depolarisation, are the practical constraints on scaling.

Limitations. The data are synthetic. Their structure (Gaussian device profiles, a single latent correlation factor, additive stealth shift) is simpler than real traffic, and the relative ranking of methods could change on captured data; the generator was designed to expose differences in difficulty between families, not to reproduce any real network. All quantum results are exact simulations, so hardware effects other than the two modelled here (gate errors that are not globally depolarising, readout error, drift) are absent. Compression to six features by PCA was imposed by the qubit budget and disadvantages the quantum model against full-feature baselines. The hyper-parameter grids are small, the number of seeds is ten, p-values are uncorrected, and the bandwidth analysis of Section VI-C uses oracle selection for the classical kernel. We did not explore other encodings, trainable kernels [18] or projected kernels. Finally, the conclusions about a replay-style attack concern the specific stealth model used here; detecting a faithful replay in practice depends on timing and protocol state that a flow-level summary does not capture.

Practical implications. For a defender with a handful of qubits and no labelled examples of a new attack, the evidence here favours a benign-only one-class model with a bandwidth chosen on benign validation data, a threshold calibrated to a tolerated false-positive rate, and a classical kernel as the first candidate. A quantum kernel is worth including as a competitor in that comparison, but nothing in this study justifies adopting it on performance grounds alone.

VIII. Conclusion

We built a controlled test bed for asking whether quantum kernels help detect rare and previously unseen attacks in industrial flow data. In simulation, a six-qubit ZZ-map SVM matched a same-input classical RBF-SVM under extreme imbalance and was more stable, and a benign-only quantum-kernel one-class SVM beat a default-bandwidth classical counterpart, but a classical kernel with a suitable bandwidth erased that advantage, and classical models using all features were stronger overall. Kernel concentration with qubit count and finite-shot noise limit how far the approach can be scaled, whereas

global depolarisation is harmless to the SVM. Future work should repeat these experiments on captured industrial traffic, add realistic hardware noise models and device runs, and include trainable and projected kernels. The released code allows each of these extensions to reuse the present protocol.

Declarations

Data and code availability. The synthetic generator, the statevector simulator, the experiment scripts and all raw result files (JSON lines) used to produce every table and figure are provided as supplementary material, with seeds fixed so that results are reproducible.

Use of AI assistance. No AI tools were used in the preparation of this manuscript.

Funding and competing interests. This study received no external funding. The authors declare no conflicts of interest.

References

[1]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.
[2]. N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in *Proc. Military Communications and Information Systems Conf. (MilCIS)*, 2015, pp. 1–6.
[3]. A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, “TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems,” *IEEE Access*, vol. 8, pp. 165130–165150, 2020.
[4]. R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 305–316.
[5]. C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[6]. V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, pp. 209–212, 2019.
[7]. M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” *Phys. Rev. Lett.*, vol. 122, no. 4, p. 040504, 2019.
[8]. Y. Liu, S. Arunachalam, and K. Temme, “A rigorous and robust quantum speed-up in supervised machine learning,” *Nat. Phys.*, vol. 17, pp. 1013–1017, 2021.
[9]. H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, “Power of data in quantum machine learning,” *Nat. Commun.*, vol. 12, p. 2631, 2021.
[10]. J. Kübler, S. Buchholz, and B. Schölkopf, “The inductive bias of quantum kernels,” in *Advances in Neural Information Processing Systems*, vol. 34, 2021.
[11]. Kaza, V., Chattu, P., Kumar, C. K., Pandraju, T. K. S., & Sirisha, U. (2026). DAVis-Net: A Dual-Attention Deep Supervision Framework for Reliable Retinal OCT Image Classification with Integrated Explainability and Uncertainty Quantification. *Mathematical and Computational Applications*, *31*(5), 190.
[12]. J. Preskill, “Quantum computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, 2018.
[13]. M. Cerezo *et al.*, “Challenges and opportunities in quantum machine learning,” *Nat. Comput. Sci.*, vol. 2, pp. 567–576, 2022.
[14]. J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, “Barren plateaus in quantum neural network training landscapes,” *Nat. Commun.*, vol. 9, p. 4812, 2018.
[15]. Klodian Dhoska, Panagiotis Kyratsis, Marek Dudek, Phani Praveen Surapaneni, and Aaron Mogeni oirere, “Smart Hybrid Intrusion Detection for IoT Networks Using Machine Learning and Neural Networks”, *BJN*, vol. 2025, pp. 197–206, Dec. 2025.
[16]. A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, “Data re-uploading for a universal quantum classifier,” *Quantum*, vol. 4, p. 226, 2020.
[17]. M. Schuld, R. Sweke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Phys. Rev. A*, vol. 103, no. 3, p. 032430, 2021.

[18]. T. Hubregtsen *et al.*, "Training quantum embedding kernels on near-term quantum computers," *Phys. Rev. A*, vol. 106, no. 4, p. 042431, 2022.
[19]. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, pp. 195–202, 2017.
[20]. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, art. 15, 2009.
[21]. B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[22]. Bikku, T., Malligunta, K. K., Thota, S., & Surapaneni, P. P. (2025). Improved quantum algorithm: a crucial stepping stone in quantum-powered drug discovery. *Journal of Electronic Materials*, *54*(5), 3434-3443.
[23]. Bikku, T., Chandolu, S. B., Praveen, S. P., Tirumalasetti, N. R., Swathi, K., & Sirisha, U. (2024). Enhancing real-time malware analysis with quantum neural networks. *Journal of Intelligent Systems and Internet of Things*, *12*(1), 57-7.
[24]. D. Shepherd and M. J. Bremner, "Temporally unstructured quantum computation," *Proc. R. Soc. A*, vol. 465, no. 2105, pp. 1413–1439, 2009.
[25]. N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems*, vol. 14, 2002.
[26]. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[27]. F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
[28]. J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
[29]. C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.
[30]. F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.