Data-Centric Evaluation of Quantum Machine Learning for Industrial Fault Detection

Kauã Dalla Riva Cucco Barbosa[1], Everson da Silva Flores[1], Bruna de Vargas Guterres[2], Silvia Silva da Costa Botelho[1] and Marcelo Rita Pias[1]

Computer Science Center. Federal University of Rio Grande (FURG), Rio Grande, Brazil[1]
Postgraduate Program on Robotics and Artificial Intelligence, Technological University of Uruguay (UTEC), Rivera, Uruguay[2]
Corresp. author email: kauadallariva@furg.br

Abstract. A data-centric evaluation of quantum machine learning (QML) is presented for industrial fault detection. A controlled experimental design is adopted. Data partitioning protocols are varied across three case studies. Preprocessing and encoding strategies are held fixed. QSVM and classical SVM are evaluated on ten datasets derived from the same sensor dataset. The results show that QSVM behavior depends on the data partitioning protocol. Classical SVM remains comparatively stable. Performance observed under contaminated partitioning is not reproduced after contamination removal. Under unsupervised conditions, both models exhibit similar performance. These results show that model behavior depends on the data partitioning protocol and should therefore be treated as a design variable in QML evaluation.

Keywords: Data-centric quantum machine learning, industrial fault detection and data partitioning

1 Introduction

Quantum machine learning (QML) integrates quantum computation into machine learning workflows to address problems that are computationally constrained in classical frameworks [8,20]. Research in this area has been oriented primarily toward circuit design, algorithm refinement and the evaluation of quantum model architectures [28,29]. Data preparation has received comparatively less attention despite its direct influence on encoding overhead, circuit depth and classification accuracy [19,3].

A model-centric perspective constrains the practical applicability of QML systems. Near-term quantum hardware (NISQ) poses limits on qubit count, coherence time and noise tolerance. All of which is likely to affect how classical data can be encoded and processed [8]. Under these constraints, the geometry of decision boundaries is shaped by the encoding strategy, which governs how data features are mapped into a Hilbert space [3,25]. Model behavior may be

modified by changes to the data representation without modifying the circuit architecture [18].

In contrast, a data-centric perspective has been proposed as an alternative to model-centric development in classical machine learning [9]. Data quality, representation and handling are treated as system design variables rather than secondary concerns. The *dc-qml framework* extends this perspective to QML systems [18]. Data ingestion, tokenization, quantum encoding and synthetic data generation are structured within a unified data-to-model workflow. Evaluation outcomes are used to guide iterative refinement of both data and model components [21].

Industrial fault detection provides a rich validation domain to examine the dc-qml framework. In manufacturing and cyber-physical systems, data are collected under operational constraints. These conditions produce datasets that are small, unbalanced and topologically complex [10,4,7]. Fault signatures appear as subtle deviations in correlated sensor signals rather than abrupt statistical changes [30,10]. Labeled fault events are rare, which limits the use of standard supervised learning approaches [4,27]. This data structure defines the conditions under which model behavior is evaluated. Performance in existing QML fault detection studies is commonly reported using aggregate metrics such as accuracy and F1 score [26,13]. The dependence of these metrics on the partitioning protocol applied to the source data is not systematically examined. This limitation calls for a controlled evaluation design. This work makes two contributions. The dc-qml workflow is introduced, in which data ingestion, tokenization, encoding and iterative refinement are organized within a unified design space. An empirical study is then carried out in which the partitioning protocol is applied to a fixed set of sensor data across three case studies. In this case, preprocessing and encoding are held constant.

The following research question is addressed in this work:

RQ: *When data partitioning protocols are varied across three case studies, with preprocessing and encoding held fixed, do QSVM and classical SVM exhibit systematic differences in classification behavior across ten industrial fault detection datasets?*

The remainder of this paper is organized as follows. Section 2 reviews related work on quantum and classical methods for fault detection. Section 3 presents the dc-qml framework and its design rationale. Section 4 describes the experimental design and evaluation setup. Section 5 reports the results across the datasets. Section 6 analyzes the effect of partitioning on model behavior. Section 7 concludes the paper.

2 Related Work

This section reviews approaches to quantum machine learning and related methods for fault detection. Differences in algorithms, datasets, feature representations and evaluation metrics are discussed. A structured summary of the reviewed studies is provided in Table 1. It can be observed that QSVM-based

models are frequently used in industrial test rigs, e.g. SCADA systems, and other relevant cyber-physical infrastructures [27]. Such applications are characterized by heterogeneous data conditions and distinct operational constraints. The recurrence of this type of model suggests a consistent preference under supervised settings. AutoQML frameworks and quantum-enhanced recurrent models are used to address multiple data modalities [21].

Table 1. Summary of Recent QML and Related Approaches for Fault Detection

Author	Algorithm	Dataset	Qubits	Features	Metrics (Best)
Sharma et al. [22]	QSVM / QSVC	3D-Printer Test Rig (Vibration Signals)	5–16	Time-domain features (RMS, Kurtosis, Skewness, Std, Median)	Acc: 96.55%
Roth et al. [21]	AutoQML (QSVM, QKRR, QGPR, QNN)	Time Series, Image, Tabular	8	PCA, UMAP, Standardization	Bal. Acc: ∼95%; Acc: ∼93%
Krishnamurthy et al. [13]	QSVM	Google Cluster	5–6	CPU, Memory, I/O, Network	Improved load balancing; reduced failure probability
Correa-Jullian et al. [5]	QSVM	WTS (SCADA)	4–32	PCA, Autoencoders	Acc: 92.5%; F1: 0.928
Badami [2]	QSVM (ZZFeatureMap)	SWaT, HAI	8	Top-8 features (Random Forest)	SWaT AUC: 0.998, F1: 0.901; HAI AUC: 0.863, F1: 0.3748
Cultice et al. [6]	QSVM	SWaT, HAI, WADI	6–8	Decision Tree, NMF, PCA	F1: 0.95 (SWaT); 0.88 (WADI); 0.86 (HAI)
Sudharson K et al. [24]	QE-LSTM	SECOM, IMMD, C-MAPSS	8	Amplitude encoding, PPCA	F1: 0.876 (SECOM); 0.898 (IMMD); RMSE: 18.1
Muandet e Schölkopf [16]	OCSMM (Classical)	SDSS / High Energy Particle Physics	N/A	5 (Particle Physics)	AUC: 0.795 (Particle Physics)
Park et al. [17]	VQOCC	Handwritten Digits / Fashion-MNIST	6–9	64–256	AUC: 100.0% / 99.4%
Tscharke et al. [27]	QSVR	KDD (and 10 real datasets)	5	5	Acc: 82.0%, F1: 0.78, Mean AUC: 0.77

2.1 QML for Industrial Fault Detection

Quantum kernel methods for industrial anomaly diagnosis have been investigated under NISQ constraints. Data preprocessing, dimensionality reduction and embedding strategies are identified as important aspects of model behavior in noisy environments [6,5]. Model performance depends on how information is structured before circuit execution.

In cyber-physical and industrial control systems, a QSVM-based supervised learning architecture has been proposed in [2]. An 8-qubit ZZFeatureMap has been developed to encode correlations between sensor nodes. Sensor relationships that are not detected by classical models are reported to be captured by this

particular encoding. A related approach is presented in [6], where classical feature engineering methods, including decision trees, PCA and NMF are used as a filtering stage. Restricting quantum encoding to reduced and dense features is shown to improve diagnostic metrics over classical baselines.

In wind turbine diagnosis, supervised detection of pitch faults is studied using SCADA data [5]. Nonlinear reduction via autoencoders is compared with linear reduction via PCA. Better accuracy is reported in a latent space of 19 features derived from autoencoders and encoded through angle-based quantum mappings. These results suggest that the encoding data input, not only the quantum circuit architecture, conditions model performance. In rotating machinery, time frequency feature extraction has been used as the primary data strategy [15,12].

A hybrid supervised model that integrates VQE and MLP is introduced in [15], in which features from vibration signals are encoded using angular gates. Entanglement topologies are applied to separate multiple fault categories. The MRMR algorithm is applied to temporal and statistical feature selection prior to QSVM classification with amplitude encoding in [12].

2.2 Classical Methods for Industrial Fault Detection

Industrial fault detection has been approached as a data classification problem across three methodological generations [14]. Physics-based modeling approaches and expert systems were developed in the earliest stage. In this case, the diagnostic coverage is bounded by the completeness of the underlying knowledge base. Signal processing methods, including wavelet decomposition and PCA, were subsequently developed to extract condition-relevant features from vibration, temperature and pressure measurements. Manually engineered features were later replaced by statistical learning approaches. SVMs have been extensively used for fault classification in rotating machinery, bearings and motors. Hierarchical representations are learned directly from raw data signals within deep neural architectures, which removes the dependence on manual feature engineering [14].

3 Data-Centric QML (dc-qml)

3.1 Industrial Data Conditions

Industrial fault detection datasets are characterized by three structural problems: class imbalance, small sample size and multi-source heterogeneity [14,1]. In equipment monitoring, for example, nominal operational data are collected continuously, whereas fault events occur infrequently. This distributional asymmetry is documented across standard benchmarks, including CWRU, IMS, and PU and biases classifiers toward the majority class when no correction is applied.

Data quality is not guaranteed even when sample volumes are appropriate. Sensor noise, calibration drift, missing values and temporal misalignment introduce errors that propagate through feature extraction and into model parameters [1]. Signal heterogeneity further complicates data representation. Industrial

measurements are created from multiple modalities, including vibration, temperature, pressure, and current, collected at different sampling rates under varying operating conditions. These conditions define the constraints that tokenization and encoding stages must address before quantum processing is applied.

3.2 dc-qml Workflow

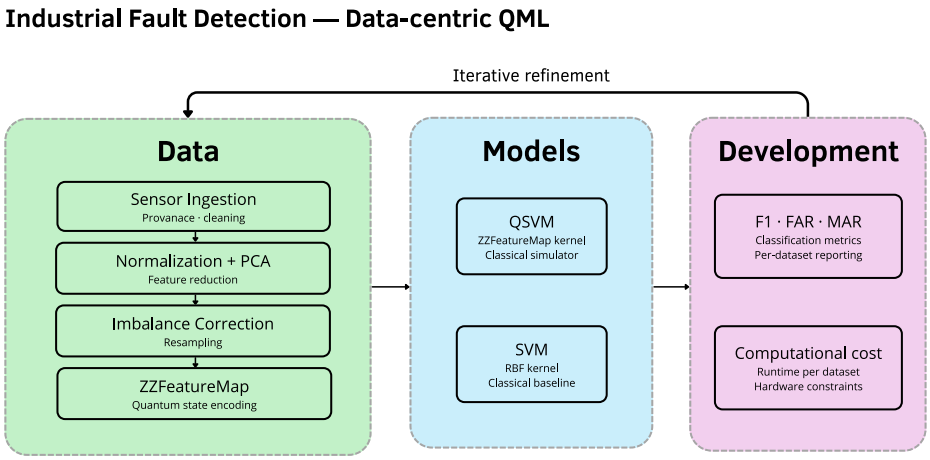

Fig. 1. The dc-qml workflow. Data are normalized, reduced and encoded via a quantum kernel. QSVM and SVM are evaluated using F1, FAR, and MAR. Results guide iterative data refinement.

The dc-qml workflow is organized into five stages, as presented in Fig. 1. Each stage addresses a data condition that arises in industrial monitoring environments, including sensor noise, class imbalance and high dimensionality. An evaluation loop connects the model outputs back to data preparation throughout the workflow. In the dc-qml methodology, *process understanding* defines how refinement is carried out across workflow stages. In traditional model-centric QML, refinement usually occurs between model development and model evaluation. In the dc-qml methodology, refinement may also return to the data phase. This return is driven by empirical observations such as low accuracy, excessive circuit depth, large qubit demand or poor robustness to noise. These observations may lead to changes in token granularity, encoding strategy, dimensionality reduction or data augmentation. This loop is methodological, it is not presented as an automatic system in which model loss directly triggers a predefined modification of token size or encoding.

Data Ingestion. Sensor data records are collected from industrial monitoring systems and data provenance is checked to confirm measurement consistency. Calibration artifacts, missing data values and temporal misalignment are identified and corrected at this stage. Dimensionality reduction and compression

strategies are applied to align high-dimensional sensor records with qubit count and circuit depth constraints [19]. In the case study reported here, features derived from an industrial system are processed at this stage before downstream encoding.

Tokenization and Segmentation. A sensor-derived feature vector is decomposed into fixed-length segments termed tokens, each mapped to one or a small group of qubits. This decomposition reduces the encoding overhead associated with high-dimensional industrial measurements. Through quantum random access codes (QRAC), n classical bits are encoded into a single qubit with success probability $p > 1/2$, which reduces circuit depth requirements under NISQ hardware constraints.

Quantum Data Encoding. Each token is mapped to a quantum state through an encoding strategy and used after tokenization. Model accuracy and circuit depth are both affected by the encoding choice [3,25]. An IQP embedding kernel is applied in the case study reported here. Feature correlations between sensor channels are captured through entangled two qubit interactions, which are appropriate for industrial fault settings where anomalies manifest as correlated deviations across multiple measurements.

Generative Synthetic Data. Fault event records are uncommon in industrial datasets, and generative models are applied to produce synthetic minority class samples. Gaussian process generators and diffusion models are used to expand the fault class before training. The generated samples are tokenized and encoded following the same procedure applied to observed data. This stage addresses the class imbalance condition documented across standard industrial benchmarks such as CWRU and SECOM.

Refinement Continuum. Model evaluation is conducted after each stage rather than only at the terminal point of the workflow. Classification metrics, including F1 score, FAR and MAR are computed per dataset and fed back into data preparation. Circuit depth, qubit count and noise sensitivity are recorded as scalability constraints alongside classification performance. When performance limits are reached, data representation is revised before model architecture is adjusted.

3.3 Problem Characterization

Supervised industrial inspection is treated here as a binary classification problem over sensor-derived feature vectors, where the target label distinguishes nominal from fault-bearing operational states. The supervised setting is adopted because the dc-qml framework is evaluated under controlled conditions where fault labels are available. Labeled fault events are rare relative to nominal observations and fault signatures manifest as subtle deviations in correlated sensor signals [14,1]. Available sample counts are substantially smaller than those used in standard deep learning benchmarks. Geometrically irregular decision boundaries are produced by these conditions in classical feature space. Quantum kernel methods are examined as an alternative, as classical data mapped into a high-dimensional

Hilbert space may reveal separations not accessible with classical kernels alone [22,5].

4 Experimental Design

Ten binary classification datasets are derived from a multivariate sensor record from the SKAB benchmark [11]. Each dataset corresponds to a distinct operating scenario, indexed by increasing geometric complexity of the decision boundary. The dc-qml preprocessing pipeline is applied uniformly across all datasets: normalization, PCA-based dimensionality reduction to four features, and IQP embedding encoding for the QSVM. The classical baseline receives the same normalized and reduced features without quantum encoding.

The role of data partitioning is examined through three case studies. Each case study fixes the preprocessing pipeline and varies only the protocol used to draw training and test samples from the time-ordered record. This isolation is required to attribute behavioral differences to data choice rather than to encoding or model adjustments.

Case Study 1: non-temporally ordered supervised split. The training set is composed of 150 normal samples drawn from the beginning of the recording and 150 anomaly samples assembled from a window centered on the changepoint. The anomaly window covers 50 samples preceding the fault and 100 samples following it, as shown in Fig. 2. The 50 pre-fault samples carry normal operating labels in the source data and are relabeled as anomaly during window construction. Temporal coherence between training samples is not preserved.

Case Study 2: temporally ordered supervised split. The training partition is composed of 156 normal samples drawn from the period immediately preceding the "changepoint", then followed by 84 fault samples starting at the changepoint, as shown in Fig. 3. The test partition is composed of 30 normal samples immediately preceding the training window and 30 fault samples immediately following the training fault block. No overlap is present between partitions, and no shuffle is applied. The normal-to-fault boundary transition is preserved in the training data, and the test set reflects the deployment condition in which a classifier encounters future observations.

Case Study 3: fully sequential unsupervised split. The training partition is composed of 240 normal samples drawn exclusively from the period preceding the changepoint, as shown in Fig. 4. No fault samples are exposed during training. The test partition is composed of 30 normal samples immediately preceding the changepoint and 30 fault samples immediately following it. A 60-sample transition window centered on the changepoint is reserved for evaluation. This protocol matches the deployment condition in which fault labels are absent and only nominal operating data are available for model fitting.

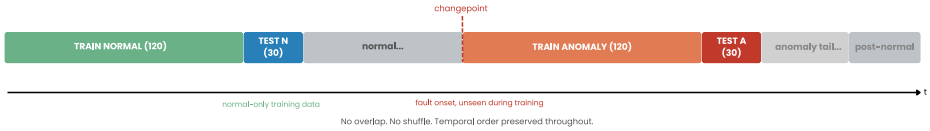

Fig. 2. Case Study 1 partitioning. The training set is composed of 150 normal samples drawn from the start of the record and 150 anomaly samples centered on the changepoint. A gap separates the end of the normal training block from the changepoint. Temporal ordering is not preserved.

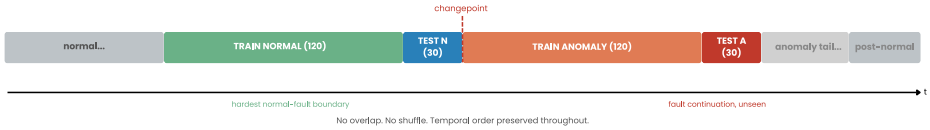

Fig. 3. Case Study 2 partitioning. Training samples are drawn contiguously across the changepoint, exposing the normal-to-fault boundary. Test samples represent the operating condition closest to fault onset and the fault continuation period.

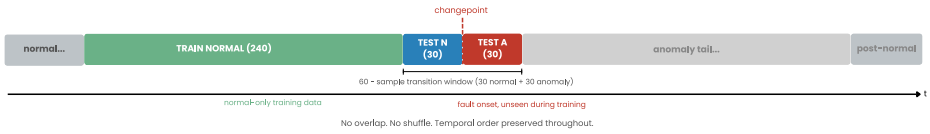

Fig. 4. Case Study 3 partitioning. Training data are restricted to nominal operating samples preceding the changepoint. The test set spans a 60 sample transition window covering the fault onset.

4.1 Evaluation Protocol

Performance is assessed using metrics sensitive to class imbalance. Aggregate accuracy is insufficient in this setting, as a classifier that predicts the majority class exclusively can achieve high accuracy without detecting any fault [23]. The distribution of classification errors across fault and nominal classes is reported separately through two complementary metrics.

The False Alarm Rate (FAR) is defined as the proportion of nominal observations incorrectly classified as faults. The Missed Alarm Rate (MAR) is defined as the proportion of fault observations incorrectly classified as nominal. Both metrics are reported alongside the F1 score. Benchmark comparability is maintained by applying the dc-qml preprocessing and encoding pipeline uniformly to both classifiers, thereby isolating the effect of the model component from that of data preparation.

4.2 Experimental Setup

The supervised QSVM has been implemented using the PennyLane framework with an IQP embedding kernel. The one-class QSVM (OC-QSVM) uses a Quantum Inversion Test (qIT) kernel implemented via Qiskit. A classical SVM with an RBF kernel and default regularization is used as the baseline. Both classifiers are trained on the same preprocessed datasets, with no access to test labels during preprocessing. Circuit depth is determined by the number of features and the number of embedding repetitions, which is fixed at two across all datasets.

The experimental protocol follows a factorial design in which factors and levels are systematically varied. Data partitioning is treated as the primary factor, with three levels corresponding to the case studies. All experimental runs are repeated seven times to assess consistency. A one-way ANOVA analysis was conducted across experimental repetitions. The results show low variability, with no statistically meaningful differences across runs, as indicated by the p-value$>$ 0.05.

The QSVM and classical SVM are evaluated on Case Studies 1 and 2, in which labeled fault samples are included in the training partition. In Case Study 3, the supervised classifiers are replaced by a one-class QSVM (OC-QSVM) and a one-class SVM (OC-SVM), both trained using nominal samples only (no labels used). The same preprocessing and encoding pipeline are executed across all three case studies. The protocol is reproducible with code and data available online.[1][2]

5 Results

F1 scores for the three case studies are presented in Fig. 5. The three panels correspond to three distinct data conditions applied to the same sensor data record. The behavior changes observed across panels can be linked to the partitioning protocol, as preprocessing and encoding are held fixed throughout.

Table 2 reports raw classification metrics for Case Study 1. Results for Case Studies 2 and 3 are reported in Table 3 and Table 4, respectively.

Case Study 1 Raw experimental results for the non-temporally ordered supervised split are presented in Table 2. The classical SVM attains an average F1 score of 0.9524 against 0.6650 for the QSVM. On dataset 9, an F1 score of 0.9677 and a FAR of 3.45% are reported for the QSVM, approaching the SVM result of 1.0000. On datasets 10, 11, 12, and 13, the QSVM achieves F1 scores below 0.60, with FAR and MAR both exceeding 30%. The SVM achieves reasonably high F1 scores on the same datasets.

[1] `https://www.dropbox.com/scl/fi/rhulsmh2guaabmheehi0j/qsvm-benchmark.zip?rlkey=y6mjf1kak3tmnrg7swvnuclbf&st=q2bylzpu&dl=0`

[2] `https://www.dropbox.com/scl/fi/4cfp9zotrjvvk0mzt1c5n/oc-qsvm-benchmark.zip?rlkey=df5ea8wx7dxja4rdv5bh7vrne&st=uui18wni&dl=0`

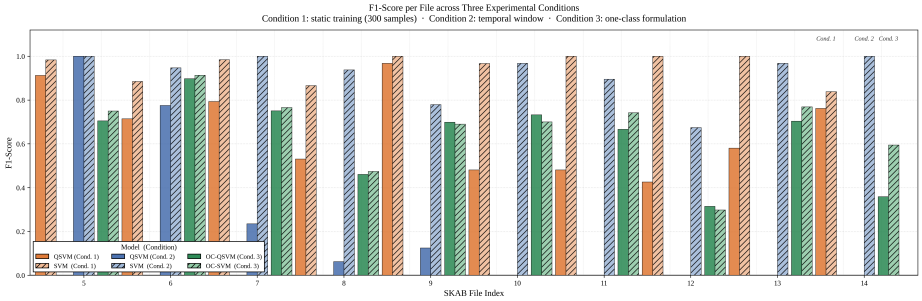

Fig. 5. F1 scores per data file across case studies. Case Study 1: non-temporally ordered supervised split. Case Study 2: temporally ordered supervised split. Case Study 3: fully sequential unsupervised split with one-class classifiers.

Table 2. Case Study 1: raw experimental results under the non-temporally ordered supervised split. FAR: False Alarm Rate. MAR: Missed Alarm Rate.

File	QSVM F1	QSVM FAR (%)	QSVM MAR (%)	SVM F1	SVM FAR (%)	SVM MAR (%)
5.csv	0.9123	**0.00**	16.13	**0.9836**	**0.00**	**3.23**
6.csv	0.7143	17.24	35.48	**0.8852**	**10.34**	**12.90**
7.csv	0.7937	24.14	19.35	**0.9841**	**3.45**	**0.00**
8.csv	0.5306	**17.24**	58.06	**0.8657**	24.14	**6.45**
9.csv	0.9677	3.45	3.23	**1.0000**	**0.00**	**0.00**
10.csv	0.4815	34.48	58.06	**0.9667**	**0.00**	**6.45**
11.csv	0.4815	34.48	58.06	**1.0000**	**0.00**	**0.00**
12.csv	0.4262	58.62	58.06	**1.0000**	**0.00**	**0.00**
13.csv	0.5806	44.83	41.94	**1.0000**	**0.00**	**0.00**
14.csv	0.7619	27.59	22.58	**0.8387**	**17.24**	**16.13**

The dispersion of QSVM F1 scores across data files is high under this protocol, ranging from 0.4262 to 0.9677. The classical SVM exhibits a narrower dispersion range with F1 scores between 0.8387 and 1.0000.

Case Study 2 Classification metrics for the temporally ordered supervised split are presented in Table 3. A bimodal pattern is observed in the QSVM column. On datasets 5 and 6, the QSVM achieves F1 scores of at least 0.78. On datasets 10 through 14, the QSVM achieves F1 scores of 0.06 or lower, with no fault detection in several files. The classical SVM achieves F1 scores of at least 0.67 across the 10 datasets. The contrast with Case Study 1 is observable in the same files. On datasets 10, 11, 12 and 13, QSVM F1 scores are below 0.60 under Case Study 1 and approach zero under Case Study 2. On dataset 9, the QSVM trend

Table 3. Case Study 2: classification metrics under the temporally ordered supervised split.

File	QSVM F1	QSVM FAR (%)	QSVM MAR (%)	SVM F1	SVM FAR (%)	SVM MAR (%)
5.csv	**1.0000**	**0.00**	**0.00**	**1.0000**	**0.00**	**0.00**
6.csv	0.7755	**0.00**	36.67	**0.9474**	**0.00**	**10.00**
7.csv	0.2353	**0.00**	86.67	**1.0000**	**0.00**	**0.00**
8.csv	0.0625	**3.33**	96.67	**0.9375**	13.33	**0.00**
9.csv	0.1250	**0.00**	93.33	**0.7792**	56.67	**0.00**
10.csv	0.0000	**0.00**	100.00	**0.9677**	6.67	**0.00**
11.csv	0.0000	**0.00**	100.00	**0.8955**	23.33	**0.00**
12.csv	0.0000	**0.00**	100.00	**0.6742**	96.67	**0.00**
13.csv	0.0000	**0.00**	100.00	**0.9677**	6.67	**0.00**
14.csv	0.0000	**0.00**	100.00	**1.0000**	0.00	**0.00**

runs in the opposite direction: F1 is 0.9677 under Case Study 1 and much lower under Case Study 2.

Case Study 3 Classification metrics for the fully sequential unsupervised split are presented in Table 4. The OC-QSVM and OC-SVM achieve comparable F1 scores across most files, ranging from 0.30 to 0.77. Neither classifier achieves the F1 levels observed for the supervised SVM in Case Study 2 on the same files. Both one-class classifiers degrade on dataset 12, with F1 scores below 0.35.

The reduced separation between OC-QSVM and OC-SVM under this protocol is consistent with the absence of fault labels during model training. The decision boundary is estimated from nominal samples alone. Quantum encoding does not produce a measurable advantage over the classical baseline in this configuration.

6 Analysis

The three case studies share a fixed preprocessing pipeline and a fixed encoding map. Differences observed across case studies can be linked to the partitioning protocol applied to the time-ordered record. Behavioral changes are therefore tied to data choice rather than to model architecture or feature engineering.

Effect of label contamination on QSVM performance The QSVM produces qualitatively different classification behavior under Case Studies 1 and 2 on the same datasets. Under Case Study 1, F1 scores range from 0.42 to 0.97 across the 10 files, with a mean of 0.6650. The 50 pre-fault samples relabeled as anomalies distort the geometry of the training set, and the resulting decision boundary is partly defined by the contaminated examples. Under Case Study

Table 4. Case Study 3: classification metrics under the fully sequential unsupervised split with one-class classifiers.

File	OC-QSVM F1	OC-QSVM FAR (%)	OC-QSVM MAR (%)	OC-SVM F1	OC-SVM FAR (%)	OC-SVM MAR (%)
5.csv	0.7048	45.13	**0.00**	**0.7500**	**35.90**	**0.00**
6.csv	0.8975	12.31	**0.00**	**0.9130**	**10.26**	**0.00**
7.csv	0.7507	26.83	**5.26**	**0.7660**	**24.39**	**5.26**
8.csv	0.4603	21.54	58.10	**0.4737**	**20.51**	**57.14**
9.csv	**0.6988**	**35.90**	10.48	0.6897	43.59	**4.76**
10.csv	**0.7326**	14.12	**31.54**	0.7000	**0.00**	46.15
11.csv	0.6667	45.95	13.04	**0.7419**	**43.24**	**0.00**
12.csv	**0.3145**	**84.76**	**44.44**	0.2985	92.86	**44.44**
13.csv	0.7038	20.47	17.65	**0.7692**	**16.28**	**11.76**
14.csv	0.3592	23.00	68.00	**0.5946**	**15.00**	**45.00**

2, in which the contamination is removed, and the temporal boundary between normal and fault samples is preserved, the QSVM collapses to F1 scores at or below 0.06 on five of the ten datasets. The classical SVM does not exhibit a comparable shift, with F1 scores remaining above 0.67 across both protocols.

The dataset 9 result shows the same effect from the opposite direction. Under Case Study 1, the QSVM achieves an F1 score of 0.9677 with a FAR of 3.45%, approaching the classical SVM result of 1.0000. Under Case Study 2, this performance is not retained. The outcome of Case Study 1 is therefore conditioned on the contaminated labels rather than on separation in the encoded feature space. Removal of the contamination exposes a kernel that does not separate under the same operating conditions under the corrected protocol.

Decision geometry under the corrected partitioning The bimodal QSVM distribution observed under Case Study 2 is tied to the topological complexity of the decision boundary in each dataset. On datasets 5 and 6, the normal-to-fault transition is captured by the IQP embedding kernel. F1 scores at or above 0.78 are observed in this case. On datasets 10 through 14, the same kernel does not separate the classes and F1 scores fall to or below 0.06. The classical SVM retains F1 scores above 0.67 on the same datasets. Sensitivity of the QSVM to the geometry produced by the corrected partitioning is therefore greater than that of the RBF kernel under identical preprocessing.

Behavior under unsupervised conditions In Case Study 3, the OC-QSVM and OC-SVM are trained only on nominal samples. The two classifiers produce F1 scores within overlapping ranges across the ten data files, with no consistent margin between them. Both classifiers degrade on dataset 12, where the transition window contains samples that are geometrically close to the nominal train-

ing distribution. The supervised QSVM under Case Study 2 does not outperform the one-class classifiers on the topologically complex datasets. Reconstruction-based decision boundaries derived from nominal data are therefore not systematically inferior to supervised quantum boundaries when the supervised training set includes the normal to fault transition.

Methodological consequence The cross-case study comparison produces three observations. QSVM classification behavior is not invariant to the partitioning protocol under fixed preprocessing. In contrast, classical SVM behavior remains comparatively stable under the same conditions. Performance metrics computed under a single partitioning protocol do not generalize to alternative protocols and may reflect data conditions rather than intrinsic model behavior. The dc-qml refinement loop treats data partitioning as a design variable rather than as a fixed experimental parameter, and provides a structured mechanism to examine this dependence.

7 Conclusions

The dc-qml methodology has been presented and examined through three case studies in industrial fault detection. Data ingestion, tokenization, quantum encoding and synthetic data generation are structured within a unified design space. Evaluation outcomes are linked back to data preparation stages through an iterative refinement loop.

The work evaluated whether changes in data-partitioning protocols lead to systematic differences in the behavior of QSVMs and classical SVMs under fixed preprocessing and encoding strategies. The results show that QSVM behavior is heavily dependent on the partitioning protocol. The classical SVM remains comparatively stable across all case studies.

Performance obtained under contaminated partitioning is not reproduced when the contamination is removed. Under unsupervised conditions, QSVM and SVM exhibit similar performance levels, with no consistent separation between the two models. These findings suggest that model behavior is not invariant to partitioning.

The research question is therefore answered in the affirmative. Systematic differences are observed, primarily driven by the partitioning protocol rather than the model architecture. Data partitioning should be treated as a design variable in QML evaluation.

Limitations

This study has three limitations. The datasets are small, and generalization to larger or different data distributions has not been established. Model outputs are not traceable to physical signal features and attribution methods for quantum kernels remain unresolved. The evaluation was conducted in a controlled environment, without accounting for factors such as sensor drift, concept drift or system integration.

Future Directions

Future work includes the development of additional partitioning protocols and encoding strategies. The sensitivity of the IQP embedding kernel should be compared with ZZFeatureMap, angle and amplitude encodings. Reconstruction-based and semi-supervised methods should be evaluated under the same protocol [23]. Further directions also include circuit compression, federated data collection and automation of the dc-qml refinement loop.

Disclosure of AI Usage

Google NotebookLM was used for summarization and Grammarly for grammar revision. All outputs were reviewed and integrated by the authors. The manuscript text reflects their own understanding and complies with institutional academic integrity policies.

References

1. Achouch, M., Dimitrova, M., Ziane, K., Sattarpanah Karganroudi, S., Dhouib, R., Ibrahim, H., Adda, M.: On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. Applied Sciences **12**(16), 8081 (Aug 2022). https://doi.org/10.3390/app12168081
2. Badami, S.: Hardware-Agnostic Quantum Kernel Feature Mapping for Anomaly Detection in Critical Infrastructure: A Cross-Testbed Validation on NISQ Processors. IEEE Access **14**, 49642–49654 (2026). https://doi.org/10.1109/ACCESS.2026.3679234
3. Benedetti, M., Lloyd, E., Sack, S.H., Fiorentini, M.: Parameterized quantum circuits as machine learning models. Quantum Science and Technology **4** (2019)
4. Choi, E., Sul, J., Kim, J.E., Hong, S.J., Gonzalez, B.I., Cembellin, P., Wang, Y.: Quantum machine learning for additive manufacturing process monitoring. Manufacturing Letters (2024)
5. Correa-Jullian, C., Cofre-Martel, S., Martín, G.S., Droguett, E.L., de Novaes Pires Leite, G., Costa, A.B.R.: Exploring quantum machine learning and feature reduction techniques for wind turbine pitch fault detection. Energies (2022)
6. Cultice, T., Onim, M.S.H., Giani, A., Thapliyal, H.: Quantum-hybrid support vector machines for anomaly detection in industrial control systems. ArXiv **abs/2506.17824** (2025)
7. Ghosh, A., Dutta, S., Das, A.K., Shukla, V.K., Moreira, F.: Quantum machine learning in industrial automation. Quantum Machine Learning in Industrial Automation (2025)
8. Hong, Y.Y., Lopez, D.J.D.: A review on quantum machine learning in applied systems and engineering. IEEE Access **13**, 144607–144631 (2025)
9. Jarrahi, M.H., et al.: The principles of data-centric ai. Commun. ACM **66**(8), 84–92 (Jul 2023). https://doi.org/10.1145/3571724
10. Jawad, A.I., Stefan-Henningsen, E., Kiani, A.: Applications of classical and quantum machine learning in manufacturing: Predictive maintenance, scheduling and tribology. Next Research (2026)

11. Katser, I.D., Kozitsin, V.O.: Skoltech anomaly benchmark (skab). https://www.kaggle.com/dsv/1693952 (2020). https://doi.org/10.34740/KAGGLE/DSV/1693952
12. Khan, M.U., Kamran, M.A., Khan, W.R.: Bearing fault diagnosis using quantum machine learning. 2023 20th International Bhurban Conference on Applied Sciences and Technology (IBCAST) pp. 1–6 (2023)
13. Krishnamurthy, B., Das, S., Shiva, S.G.: Early detection of virtual machine failures in cloud computing using quantum-enhanced support vector machine. Mathematics (2026)
14. Lei, L., Li, W., Zhang, S., Wu, C., Yu, H.: Research Progress on Data-Driven Industrial Fault Diagnosis Methods. Sensors **25**(9), 2952 (May 2025). https://doi.org/10.3390/s25092952
15. Maior, C.B.S., Araújo, L.M.M., Lins, I.D., Moura, M.D.C., Droguett, E.L.: Prognostics and health management of rotating machinery via quantum machine learning. IEEE Access **11**, 25132–25151 (2023)
16. Muandet, K., Scholkopf, B.: One-class support measure machines for group anomaly detection. ArXiv **abs/1303.0309** (2013)
17. Park, G., Huh, J., Park, D.K.: Variational quantum one-class classifier. Machine Learning: Science and Technology **4** (2022)
18. Pere, C.: Data complexity: a threshold between classical and quantum machine learning – part i (2025)
19. Raisa, M., et al.: A Parallel Quantum Feature Encoding Scheme for Effective Classical Data Classification in Quantum Convolutional Neural Networks. Proc. IEEE TENCON (2023). https://doi.org/10.1109/TENCON58879.2023.10322543
20. Rodríguez-Díaz, F., Gutiérrez-Avilés, D., Troncoso, A., Martínez-Álvarez, F.: A survey of quantum machine learning: Foundations, algorithms, frameworks, data and applications. ACM Computing Surveys (2025)
21. Roth, M., Kreplin, D.A., Basilewitsch, D., Bravo, J.F., Klau, D., Marinov, M., Pranjic, D., Stühler, H., Willmann, M., Zöller, M.A.: Autoqml: A framework for automated quantum machine learning. 2025 IEEE International Conference on Quantum Software (QSW) pp. 81–91 (2025)
22. Sharma, V., Gupta, S., Mehta, G., Lad, B.K.: A quantum-based diagnostics approach for additive manufacturing machine. IET Collaborative Intelligent Manufacturing (2021)
23. Shukla, V., Shukla, A., S. K., S.P., Shukla, S.: A systematic survey: Role of deep learning-based image anomaly detection in industrial inspection contexts. Frontiers in Robotics and AI **12**, 1554196 (Jun 2025). https://doi.org/10.3389/frobt.2025.1554196
24. Sudharson, K., Varsha, S., Santhiya, R., Rajalakshmi, D.: Quantum-enhanced lstm for predictive maintenance in industrial iot systems. MethodsX **15** (2025)
25. Thanasilp, S., et al.: Exponential concentration in quantum kernel methods. Nature Communications **15**(1), 5200 (Jun 2024). https://doi.org/10.1038/s41467-024-49287-w
26. Tomar, S., Tripathi, R., Kumar, S.: Comprehensive survey of qml: From data analysis to algorithmic advancements. ArXiv **abs/2501.09528** (2025)
27. Tscharke, K., Issel, S., Debus, P.: Semisupervised anomaly detection using support vector regression with quantum kernel. 2023 IEEE International Conference on Quantum Computing and Engineering (QCE) **01**, 611–620 (2023)
28. Yang, H., Li, X., Liu, Z., Pedrycz, W.: Improved differential privacy noise mechanism in quantum machine learning. IEEE Access **11**, 50157–50164 (2023)

29. Zaman, K., Marchisio, A., Hanif, M.A., Shafique, M.: A survey on quantum machine learning: Current trends, challenges, opportunities, and the road ahead **abs/2310.10315** (2023)
30. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., Chawla, N.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: AAAI Conference on Artificial Intelligence (2018)