future internet

MDPI

Article

FedRBF-SVM: A Federated Support Vector Machine for IIoT-Based Anomaly Detection in Upstream Oil and Gas Wells

M. Baqer

Department of Computer Engineering, College of Information Technology, University of Bahrain, Zallaq P.O. Box 32038, Bahrain; mbaqer@uob.edu.bh

Abstract

Abnormal operating conditions and equipment malfunctions in upstream oil and gas wells can compromise safety, disrupt production, and cause substantial economic losses. This study proposes FedRBF-SVM, a federated radial basis function (RBF) support vector machine (SVM) approach for collaborative anomaly classification across Industrial Internet of Things (IIoT)-enabled wells. Each participating well, represented as a client, trains and retains a local RBF-SVM using labeled samples and returns only a scalar decision-function score for each inference sample. The server averages these scores and applies a validation-calibrated decision threshold. The approach was evaluated on the 3W oil-well dataset in a simulated federated setting using different numbers of locally stored training patterns, three target false-positive rate (FPR) operating points, and a centralized RBF-SVM reference. FedRBF-SVM performance generally improved as the number of locally stored training patterns increased. With 2500 training patterns per client and a target FPR of 0.05, FedRBF-SVM achieved an F_1-score of approximately 0.79, a precision of approximately 0.97, and a recall of approximately 0.67. These results demonstrate the feasibility of federated SVM for distributed oil-well monitoring while keeping training data and models at the clients.

Keywords: upstream oil and gas; federated learning; support vector machine; anomaly detection; distributed well monitoring; Industrial Internet of Things (IIoT); Internet of Thigs (IoT)

1. Introduction

Upstream oil and gas activities, particularly well operations, are often conducted in remote locations under highly dynamic and uncertain operating conditions. Deviations from normal operating behavior can result in production losses, equipment damage, safety incidents, and costly downtime. Modern well operations continuously generate large volumes of multivariate sensor data, including pressure, temperature, flow-rate, and vibration measurement [1–5]. The volume and complexity of these data make manual monitoring impractical and motivate the development of automated data-driven anomaly-detection methods [6–8].

Industry 4.0 technologies are reshaping upstream oil and gas operations [9,10]. In parallel, machine learning has become increasingly important throughout the upstream oil and gas sector, supporting applications such as reservoir characterization, drilling optimization, production forecasting, equipment condition monitoring, and anomaly detection [11]. Industrial Internet of Things (IIoT) architectures deployed across drilling rigs, wellheads, flowlines, and surface facilities incorporate distributed sensor networks that acquire high-frequency measurements. These networks support condition-based monitoring, early fault

Check for updates

Academic Editors: Ivan Cibrario Bertolotti and Flavio Lombardi

Received: 13 August 2026
Revised: 30 September 2026
Accepted: 1 October 2026
Published: 5 October 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

detection, and predictive maintenance [12]. Moreover, cloud- and edge-computing architectures enable the processing of terabyte-scale seismic datasets and real-time drilling data streams, thereby facilitating field-wide monitoring and supporting near-real-time decision-making beyond the capabilities of conventional centralized, batch-oriented analysis [10]. Furthermore, this infrastructure supports the increasing adoption of machine-to-machine (M2M) communication and collaborative intelligence. Federated learning can further advance autonomous upstream oil and gas operations. Collectively, these technologies can strengthen data-driven coordination across distributed assets while improving operational productivity, efficiency, and safety [13–17].

A key challenge in upstream oil and gas operations is the detection of abnormal operating conditions or events, commonly referred to as anomalies. Anomaly detection involves identifying patterns that deviate from expected normal behavior, and its formulation depends strongly on the availability of labeled abnormal-event data [18–20]. When anomaly labels are unavailable or extremely scarce, one-class and unsupervised approaches are commonly used to characterize normal operating behavior and identify deviations from it [6,21]. In contrast, when labeled normal and abnormal samples are available, supervised binary classifiers can exploit information from both classes to construct discriminative decision boundaries. This study adopts the supervised setting and formulates oil and gas well anomaly detection as a binary classification problem using operating patterns from the 3W dataset [22].

Anomaly-detection problems frequently involve unequal class distributions, although the direction and magnitude of the imbalance depend on dataset construction and preprocessing. In operational settings, undesirable events occur less frequently than periods of normal operation. However, abnormal windows constitute the majority of the processed window-level dataset evaluated in this study because all nonzero 3W event classes are combined into a single abnormal class. In addition to unequal class proportions, abnormal patterns may resemble normal operating behavior, operating conditions may evolve over time, and measurement noise may exhibit characteristics similar to genuine anomalies. These factors increase the likelihood of missed detections and false alarms [18,23]. Such challenges motivate the use of supervised classifiers such as support vector machines (SVMs), particularly in distributed industrial environments, where labeled data may be available locally but cannot be shared across organizational boundaries or transferred to a centralized server for model training and analysis.

Anomaly detection in upstream oil and gas has evolved from early sensor-driven approaches to more recent data-centric studies of well production time series. Martí et al. introduced one of the earliest petroleum-industry approaches that combined a one-class SVM with segmentation for sensor-based anomaly detection using oil-platform data [21]. Subsequently, Wong et al. [24] investigated sensor abnormality detection on oil and gas platforms. Other studies have focused on multivariate well datasets: Vargas et al. [22,25] provided a public benchmark containing rare undesirable events in oil wells, Fernandes et al. [6] compared one-class classifiers using multivariate well time series, and Alharbi et al. [26] proposed explainable and interpretable anomaly-detection models for production data. Together, these studies reflect a shift toward data-driven learning from operational sensor streams that reduces reliance on rule-based monitoring. This transition is particularly relevant because operational sensor data are high-dimensional, noisy, and continuously generated.

However, existing research has several limitations. Earlier petroleum-industry studies are often constrained by small-scale or platform-specific sensor setups, which may limit their generalizability across fields and operating conditions, and several methods depend on carefully designed preprocessing or segmentation steps that may be difficult to maintain in practice [21,24]. Benchmark-driven multivariate studies remain constrained by the scarcity

of rare-event samples, class imbalance, and the difficulty of validating models under realistic operating variability. In addition, many approaches are evaluated offline rather than under real-time operating conditions [6,22,25,26]. These limitations motivate the development of supervised anomaly-detection methods that can operate effectively under unequal class distributions and limited local training data.

Federated learning offers a data-locality-preserving alternative to classical centralized detection by enabling collaborative learning and decision-making across distributed clients without requiring their raw local datasets to be centralized [13,15,16]. This paradigm is especially relevant to oil and gas operations, where well data may be distributed across geographically dispersed assets and managed across multiple organizations. However, local datasets can differ substantially because of field-specific operating conditions, production regimes, sensor characteristics, and equipment configurations, which complicates collaborative learning across sites. Federated learning has nonetheless shown promise in industrial and geoscience applications with data-sharing constraints, including productivity prediction and other petroleum-engineering tasks [27–29].

Despite this progress, the integration of federated learning with SVM-based anomaly detection for oil and gas wells remains relatively underexplored. Although Chen et al. [29] evaluated a federated approach on the 3W dataset, they used deep models rather than kernel classifiers. Recent studies of federated and distributed anomaly detection have shown that SVM-based approaches can be adapted to distributed environments, providing a computationally efficient and relatively interpretable alternative to more complex deep-learning approaches, particularly when training data are limited and communication efficiency is important [23,30,31]. These characteristics are especially relevant to oil-well anomaly detection, where limited and unequally distributed training data may be dispersed across geographically separated wells. In such environments, collaborative analysis that does not require transferring complete local operational datasets to a central server is desirable.

Motivated by these challenges, this study proposes FedRBF-SVM, a federated radial basis function (RBF) SVM approach for anomaly detection in oil and gas wells. The primary objective is to enable collaborative anomaly classification across geographically dispersed well sites while preserving data locality. Each site trains an RBF-SVM classifier on its historical operational data to identify abnormal operating states. In contrast to fully centralized learning architectures, the proposed approach is designed for distributed environments in which labeled operational data are retained locally and may be impractical to centralize. The decision-score aggregation mechanism restricts each client-to-server response to a scalar decision-function score, while local training data and locally trained RBF-SVM models remain at their respective clients.

The contributions of this study are threefold. First, the study formulates oil and gas well anomaly detection as a distributed classification problem suitable for operational data retained locally by multiple clients. Second, FedRBF-SVM enables collaborative anomaly classification through decision-score aggregation without exchanging raw well data or locally trained models. Third, the approach is evaluated in a simulated distributed setting at validation-calibrated false-alarm operating points, with particular attention to class imbalance, varying levels of local training-data availability, and performance metrics appropriate for anomaly detection, and a benchmark comparison separates the effect of decision-score aggregation from that of aggregate training-data availability.

The remainder of this paper is organized as follows. Section 2 presents the background and key concepts underlying the proposed approach. Section 3 describes the proposed FedRBF-SVM approach in detail. Section 4 presents the evaluation framework and the performance metrics and analyzes the communication and computational costs. Section 5 describes the dataset and simulation setup and presents the performance results. Section 6

discusses the results, and Section 7 concludes the paper and outlines directions for future research.

2. Background

SVMs have demonstrated strong performance in fault detection and anomaly classification tasks involving multivariate sensor data in industrial settings [32], including upstream oil and gas operations [6,22,25]. However, the centralized training paradigm used by conventional SVMs presents significant limitations in real-world upstream deployments, where sensor data are generated across geographically dispersed wells and may be managed by multiple independent operators. Centralized data aggregation can therefore be impractical because of data privacy requirements, regulatory and competitive constraints, and communication bandwidth limitations [33]. Moreover, transmitting raw well data to a central server may introduce additional cybersecurity risks, particularly in critical infrastructure environments [34]. These challenges motivate distributed learning approaches that preserve the locality of operational training data while enabling collaborative anomaly classification.

When operational data cannot readily leave the sites where they are generated, federated learning offers a way to train models jointly without pooling raw measurements [13,35–38]. Its decentralized architecture and preservation of data locality have made it an active research area since it was first proposed [39]. In conventional federated learning, clients optimize a shared global model while their training data remain local. Each client performs local optimization and transmits model parameters or updates to a server [13,33,35,40], which aggregates them to refine the global model over successive communication rounds. However, data locality should not be interpreted as a formal privacy guarantee because information may still be inferred from the transmitted model updates.

The following subsections introduce the main concepts underlying the proposed approach, namely: the SVM, the classical RBF-SVM, and federated learning.

2.1. Support Vector Machine Algorithm

An SVM is a supervised learning algorithm rooted in statistical learning theory; its soft-margin formulation for binary classification was introduced by Cortes and Vapnik [41]. The SVM seeks the separating hyperplane with the largest margin between the two classes while penalizing margin violations and classification errors. Given a training dataset $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ denotes an input feature vector and $y_i \in \{-1, +1\}$ the corresponding class label, the soft-margin SVM solves the following constrained optimization problem:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{1}$$

subject to

$$y_i\left(w^\top x_i + b\right) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \qquad i = 1, \ldots, n, \tag{2}$$

where w denotes the weight vector, b denotes the bias term, ξ_i denotes the slack variable associated with training sample i that permits margin violations, including misclassifications, and $C > 0$ denotes the regularization parameter that controls the trade-off between maximizing the margin and penalizing classification errors. The real-valued decision function is defined as

$$g(x) = w^\top x + b. \tag{3}$$

The corresponding predicted class label is

$$\hat{y} = \text{sign}(g(x)). \tag{4}$$

To address nonlinear classification problems, an SVM can employ a kernel function associated with a feature mapping from the input space to a feature space $\mathcal{H}$, defined as $\phi : \mathbb{R}^d \to \mathcal{H}$. Rather than explicitly computing the mapped feature vectors, the kernel function directly evaluates their inner product in $\mathcal{H}$ as

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}. \tag{5}$$

This study adopts the RBF kernel, which is described in detail in the following subsection. The corresponding kernelized soft-margin SVM dual problem is

$$\max_{\alpha} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right], \tag{6}$$

subject to

$$\begin{aligned} 0 \le \alpha_i \le C, \quad i = 1, \ldots, n, \\ \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{7}$$

Here, α_i denotes the Lagrange multiplier associated with the ith training sample, and α^* denotes the optimal dual solution. In the exact dual solution, samples for which $\alpha_i^* > 0$ are support vectors and determine the decision boundary. In numerical implementations, a small positive tolerance ϵ is commonly used to identify nonzero dual coefficients, such that a sample is treated as a support vector when $\alpha_i^* > \epsilon$.

2.2. RBF-SVM Algorithm

Among the kernel functions commonly used with SVMs, the RBF kernel is widely adopted because of its ability to model complex nonlinear decision boundaries. Algorithm 1 presents the classical RBF-SVM training and classification procedure. The RBF kernel is defined as

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \qquad \gamma > 0. \tag{8}$$

Here, $\|x_i - x_j\|^2$ denotes the squared Euclidean distance between the two input vectors. The kernel parameter γ controls the spatial extent of the Gaussian basis function and can be expressed as $\gamma = 1/(2\ell^2)$, where ℓ denotes the kernel length scale.

Let γ_{cfg} denote the selected RBF kernel setting, which is either a positive numerical value or the heuristic option `scale`, as implemented in scikit-learn [42]. When $\gamma_{\text{cfg}} =$ `scale` is selected, it is converted into an effective numerical coefficient γ before model training. For the centralized RBF-SVM, this coefficient is computed from the standardized training matrix as

$$\gamma = \frac{1}{d \ \mathrm{Var}(\tilde{X})}, \tag{9}$$

where $\tilde{X} \in \mathbb{R}^{n \times d}$ denotes the standardized training matrix, d the number of input features, and $\mathrm{Var}(\tilde{X})$ the population variance computed over all entries of the matrix. When a positive numerical value is selected directly, the effective coefficient is set to $\gamma = \gamma_{\text{cfg}}$.

The RBF kernel is a translation-invariant similarity function whose value depends only on the distance between two samples. For $\gamma > 0$, it is positive definite; hence, the associated kernel matrix is positive semidefinite, and the SVM dual problem remains a convex quadratic programming problem.

Algorithm 1: Classical RBF-SVM training and classification

Input: Standardized training dataset $\tilde{\mathcal{D}} = \{(\tilde{x}_i, y_i)\}_{i=1}^{n}$, where $\tilde{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$; regularization parameter $C > 0$; effective numerical RBF coefficient $\gamma > 0$; support-vector tolerance $\epsilon > 0$; standardized test sample $\tilde{x}_{\text{test}} \in \mathbb{R}^d$

Output: Predicted class label $\hat{y} \in \{-1, +1\}$

`/* - Kernel-Matrix Construction - */`

Initialize the kernel matrix $K \in \mathbb{R}^{n \times n}$;

for $i = 1$ **to** n **do**

  for $j = 1$ **to** n **do**

    $K_{ij} \leftarrow \exp\left(-\gamma \|\tilde{x}_i - \tilde{x}_j\|^2\right)$;

`/* - SVM Dual Optimization - */`

Solve the soft-margin SVM dual problem in Equations (6) and (7), using the standardized training dataset, to obtain α^*;

`/* - Support-Vector Identification - */`

Identify the support-vector set

$\mathcal{S} \leftarrow \{i \mid \alpha_i^* > \epsilon\}$;

Identify the margin support-vector set

$\mathcal{S}^\circ \leftarrow \{i \mid \epsilon < \alpha_i^* < C - \epsilon\}$;

`/* - Bias Estimation - */`

if $\mathcal{S}^\circ \neq \varnothing$ **then**

  Estimate the bias term using the margin support vectors

  $$b \leftarrow \frac{1}{|\mathcal{S}^\circ|} \sum_{r \in \mathcal{S}^\circ} \left[y_r - \sum_{i \in \mathcal{S}} \alpha_i^* y_i \exp\left(-\gamma \|\tilde{x}_i - \tilde{x}_r\|^2\right) \right];$$

else

  Retain the intercept returned by the underlying SVM solver;

`/* - Decision-Function Evaluation - */`

Compute the real-valued decision score

$$g(\tilde{x}_{\text{test}}) \leftarrow \sum_{i \in \mathcal{S}} \alpha_i^* y_i \exp\left(-\gamma \|\tilde{x}_i - \tilde{x}_{\text{test}}\|^2\right) + b;$$

`/* - Binary Classification - */`

if $g(\tilde{x}_{\text{test}}) \geq 0$ **then**

  $\hat{y} \leftarrow +1$;

else

  $\hat{y} \leftarrow -1$;

return $\hat{y}$

Model behavior depends jointly on γ and C. Because the kernel response $K(x, x_i)$ decays exponentially with $\|x - x_i\|^2$, the value of γ sets the distance over which each support vector affects the decision score at a query point x. When γ is large, this effect is confined to the immediate neighborhood of each support vector, so the decision function can follow fine-grained structure in the training data and becomes prone to overfitting. When γ is small, each support vector contributes over a wider region, the decision function varies more gradually, and underfitting becomes the main risk. The parameter C weights the slack penalty in Equation (1) against the margin term. Increasing C makes margin violations more costly, so the solver tends to reduce training errors at the cost of a narrower margin and a more complex model; decreasing C tolerates more violations in exchange for a wider margin. Because the two parameters interact, they jointly govern the bias–variance trade-off of the RBF-SVM and are usually tuned together by grid search with cross-validation [43].

The real-valued decision score of the RBF-SVM in dual form is expressed as

$$g(x) = \sum_{i=1}^{n} \alpha_i^* y_i K(x, x_i) + b = \sum_{i \in \mathcal{S}} \alpha_i^* y_i \exp\Big(-\gamma \|x - x_i\|^2\Big) + b, \tag{10}$$

where $\mathcal{S} = \{i \mid \alpha_i^* > \epsilon\}$ denotes the support-vector set identified using the numerical tolerance $\epsilon > 0$. The predicted class label is then obtained by applying a zero decision threshold:

$$\hat{y} = \begin{cases} +1, & g(x) \geq 0, \\ -1, & g(x) < 0. \end{cases} \tag{11}$$

The resulting decision boundary corresponds to the zero-level set of a weighted combination of RBF kernel evaluations centered on the support vectors, with γ governing the spatial scale of their influence. This formulation enables an RBF-SVM to represent flexible nonlinear decision boundaries while preserving the convexity of the underlying SVM optimization problem for fixed values of C and γ. These properties motivate the use of RBF-SVM models in this study.

For consistency with the experimental implementation, Algorithm 1 assumes that the training and test feature vectors have been standardized using parameters estimated exclusively from the training data, and it uses the corresponding effective numerical RBF coefficient $\gamma > 0$.

2.3. Federated Learning

Federated Averaging (FedAvg) is one of the foundational federated learning algorithms [33]. It enables multiple clients to collaboratively train a shared global model while retaining their raw training data locally. Rather than requiring all clients to participate in every communication round, FedAvg may select a subset of clients according to a participation fraction $C_f \in (0, 1]$, thereby reducing the communication and computational costs per round.

In communication round t, the server broadcasts the current global model parameters ω_t to the selected client subset $\mathcal{C}_t$. Each participating client i initializes its local model with ω_t and performs E epochs of local optimization on its local dataset $\mathcal{D}_i$. For a minibatch $\mathcal{B} \subseteq \mathcal{D}_i$, a local stochastic gradient update can be expressed as

$$\omega_{s+1}^{(i)} \leftarrow \omega_s^{(i)} - \eta \nabla f_i\Big(\omega_s^{(i)}; \mathcal{B}\Big), \tag{12}$$

where s denotes the local optimization step, $\eta > 0$ denotes the learning rate, and f_i denotes the local objective function of client i.

After local training, each participating client returns its updated model parameters $\omega_{t+1}^{(i)}$ to the server. The server then performs sample-size-weighted aggregation as follows:

$$\omega_{t+1} \leftarrow \sum_{i \in \mathcal{C}_t} \frac{n_i}{\sum_{j \in \mathcal{C}_t} n_j} \omega_{t+1}^{(i)}, \tag{13}$$

where $n_i = |\mathcal{D}_i|$ denotes the number of local training samples held by client i. This weighting ensures that clients with larger local datasets contribute proportionally more to the aggregated global model. The resulting ω_{t+1} is then broadcast to the clients selected for the next communication round. The procedure is repeated until T communication rounds have been completed or another predefined stopping criterion has been met. The final global model parameters are denoted by ω^*. Algorithm 2 presents the FedAvg procedure.

Algorithm 2: Federated Averaging (FedAvg)

Input: Number of clients K; local datasets $\{\mathcal{D}_i\}_{i=1}^{K}$, where $n_i = |\mathcal{D}_i|$; number of communication rounds T; client participation fraction $C_f \in (0, 1]$; number of local epochs E; minibatch size B; learning rate $\eta > 0$

Output: Final global model parameters ω^*

`/* - Server Processing -` */

Initialize the global model parameters ω_0;

for $t = 0, 1, \ldots, T-1$ **do**

  Select a participating-client subset $\mathcal{C}_t \subseteq \{1, \ldots, K\}$ such that $|\mathcal{C}_t| = \max\left(\left\lfloor C_f K \right\rfloor, 1\right)$;

  for *each client* $i \in \mathcal{C}_t$ *in parallel* **do**

    $\omega_{t+1}^{(i)} \leftarrow \textsc{ClientUpdate}(i, \omega_t, E, B, \eta)$;

  Compute the total number of training samples held by the participating clients $n_{\mathcal{C}_t} \leftarrow \sum_{i \in \mathcal{C}_t} n_i$;

  Aggregate the updated local model parameters $\omega_{t+1} \leftarrow \sum_{i \in \mathcal{C}_t} \frac{n_i}{n_{\mathcal{C}_t}} \omega_{t+1}^{(i)}$;

$\omega^* \leftarrow \omega_T$;

return ω^*

`/* - Client Processing -` */

Procedure *ClientUpdate* (i, ω, E, B, η)**:**

  $\omega_{\text{loc}} \leftarrow \omega$;

  for $e = 1, 2, \ldots, E$ **do**

    Shuffle $\mathcal{D}_i$ and partition it into minibatches of size B;

    for *each minibatch* $\mathcal{B}$ **do**

      $\omega_{\text{loc}} \leftarrow \omega_{\text{loc}} - \eta \nabla f_i(\omega_{\text{loc}}; \mathcal{B})$;

  return ω_{loc} *to the server*

3. FedRBF-SVM Approach

This section presents the FedRBF-SVM approach for collaborative anomaly classification in distributed environments. The approach extends the classification capability of the classical RBF-SVM to settings in which operational data are retained locally by multiple clients and may be impractical to centralize because of privacy, ownership, or communication constraints. Unlike a centralized RBF-SVM, which requires access to the complete training dataset to construct a single kernel model, FedRBF-SVM allows each participating client to independently train and retain a local RBF-SVM.

All clients use the same globally selected SVM regularization parameter C_{svm} and RBF kernel setting γ_{cfg}. When γ_{cfg} is a positive numerical value, each client uses $\gamma_i = \gamma_{\text{cfg}}$; when $\gamma_{\text{cfg}} =$ `scale`, each client computes its effective coefficient γ_i from its locally standardized training data. Thus, the kernel setting is common across clients, whereas the effective numerical coefficient may be client-specific.

3.1. Local RBF-SVM Model Training

FedRBF-SVM is intended for distributed deployment across geographically dispersed oil and gas wells. Each participating well is assumed to host a client with local processing, storage, and communication capabilities. Client i maintains a local dataset

$$\mathcal{D}_i = \left\{ \left(x_j^{(i)}, y_j^{(i)} \right) \right\}_{j=1}^{n_i}, \tag{14}$$

where $x_j^{(i)} \in \mathbb{R}^d$ denotes the jth feature vector held by client i, $y_j^{(i)} \in \{-1, +1\}$ the corresponding class label, and n_i the number of training samples in the local dataset.

Before training its local model, each client independently estimates its feature-standardization parameters from its local training data. Let μ_i and σ_i denote the client-specific vectors of feature-wise means and scaling factors, respectively. For each feature, the scaling factor equals its locally estimated standard deviation when that standard deviation is positive. A scaling factor of 1 is used when the feature is locally constant. Each local training vector is then standardized as

$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \mu_i}{\sigma_i}, \tag{15}$$

where subtraction and division are applied element-wise. Under this convention, a locally constant feature is mapped to zero. The parameters μ_i and σ_i are retained at client i and applied unchanged to the validation and test samples evaluated by that client, so no information from the validation or test sets is used in standardization.

Let γ_i denote the effective numerical RBF coefficient used by client i. Consistent with Equation (9), it is determined as

$$\gamma_i = \begin{cases} \dfrac{1}{d \ \mathrm{Var}(\tilde{X}_i)}, & \gamma_{\mathrm{cfg}} = \texttt{scale}, \\ \gamma_{\mathrm{cfg}}, & \gamma_{\mathrm{cfg}} \in \mathbb{R}_{>0}, \end{cases} \tag{16}$$

where $\tilde{X}_i \in \mathbb{R}^{n_i \times d}$ denotes client i's standardized local training matrix, and $\mathrm{Var}(\tilde{X}_i)$ the population variance computed over all entries of this matrix.

Because feature-wise standardization gives each nonconstant feature zero mean and unit variance, the overall population variance $\mathrm{Var}(\tilde{X}_i)$ is typically close to one. Consequently, when every retained feature has nonzero variance, the `scale` setting produces $\gamma_i \approx 1/d$. However, γ_i may differ across clients if their standardized local training matrices have different overall variances, for example, because some features are constant within particular local datasets. Using the standardized local training vectors, client i constructs its local RBF kernel matrix as

$$K_{jk}^{(i)} = \exp\left(-\gamma_i \left\| \tilde{x}_j^{(i)} - \tilde{x}_k^{(i)} \right\|^2\right), \qquad j, k = 1, \ldots, n_i. \tag{17}$$

Client i then solves the standard soft-margin SVM dual optimization problem

$$\max_{\alpha^{(i)}} \left[\sum_{j=1}^{n_i} \alpha_j^{(i)} - \frac{1}{2} \sum_{j=1}^{n_i} \sum_{k=1}^{n_i} \alpha_j^{(i)} \alpha_k^{(i)} y_j^{(i)} y_k^{(i)} K_{jk}^{(i)} \right], \tag{18}$$

subject to

$$0 \le \alpha_j^{(i)} \le C_{\mathrm{svm}}, \qquad j = 1, \ldots, n_i, \tag{19}$$

and

$$\sum_{j=1}^{n_i} \alpha_j^{(i)} y_j^{(i)} = 0. \tag{20}$$

Here, $C_{\mathrm{svm}} > 0$ denotes the globally selected SVM regularization parameter, which is applied to all participating clients and controls the trade-off between maximizing the margin and penalizing margin violations and classification errors.

After solving the local dual optimization problem, client i identifies its support-vector set as

$$\mathcal{S}_i = \left\{ j \,\middle|\, \alpha_j^{(i)*} > \epsilon \right\}, \tag{21}$$

where $\epsilon > 0$ denotes a small numerical tolerance used to treat numerically insignificant dual coefficients as zero. The corresponding margin support-vector set is defined as

$$\mathcal{S}_i^{\circ} = \left\{ j \,\middle|\, \epsilon < \alpha_j^{(i)*} < C_{\text{svm}} - \epsilon \right\}. \tag{22}$$

When $\mathcal{S}_i^{\circ}$ is nonempty, the local bias term is estimated as

$$b_i = \frac{1}{|\mathcal{S}_i^{\circ}|} \sum_{r \in \mathcal{S}_i^{\circ}} \left[y_r^{(i)} - \sum_{j \in \mathcal{S}_i} \alpha_j^{(i)*} y_j^{(i)} \exp\left(-\gamma_i \left\| \tilde{x}_j^{(i)} - \tilde{x}_r^{(i)} \right\|^2 \right) \right]. \tag{23}$$

If $\mathcal{S}_i^{\circ}$ is empty under the selected numerical tolerance, the bias term provided by the underlying local SVM implementation is retained. This avoids estimating the bias term from bounded support vectors, for which the Karush–Kuhn–Tucker margin-equality condition need not hold. The locally trained support-vector model is represented by

$$\mathcal{M}_i = \left\{ \left(\tilde{x}_j^{(i)}, y_j^{(i)}, \alpha_j^{(i)*} \right) \,\middle|\, j \in \mathcal{S}_i \right\}. \tag{24}$$

The resulting local model state at client i consists of $\mathcal{M}_i$, the bias term b_i, the effective RBF coefficient γ_i, and the feature-standardization parameters μ_i and σ_i. These quantities are retained locally and are not transmitted to the coordinating server. Although C_{svm} and γ_{cfg} are common across clients, the preprocessing parameters (and, when $\gamma_{\text{cfg}} = \texttt{scale}$, the coefficient γ_i) are determined independently from each client's local training data. Algorithm 3 summarizes the local RBF-SVM training procedure.

3.2. FedRBF-SVM Federated Inference

After local training, FedRBF-SVM performs collaborative inference without transmitting the locally trained RBF-SVM models to the coordinating server. For each inference request, a subset of clients is selected to participate in the prediction, and the input feature vector $x \in \mathbb{R}^d$ is made available to each of them. Each selected client evaluates the sample using its locally retained model and returns only the resulting scalar decision-function score to the server. The server then aggregates the received scores and applies a calibrated decision threshold to obtain the final binary prediction.

Let K denote the total number of clients, and let $\mathcal{C} \subseteq \{1, \ldots, K\}$ denote the set of clients selected for the current inference request. Given the client participation fraction $C_f \in (0, 1]$, the number of participating clients is

$$M = |\mathcal{C}| = \max\left(\left\lfloor C_f K \right\rfloor, 1 \right). \tag{25}$$

For an input sample x, each participating client $i \in \mathcal{C}$ first standardizes the sample using the client-specific preprocessing parameters retained during local training:

$$\tilde{x}^{(i)} = \frac{x - \mu_i}{\sigma_i}. \tag{26}$$

The locally retained RBF-SVM at client i then computes the scalar decision-function score

$$s_i(x) = \sum_{j \in \mathcal{S}_i} \alpha_j^{(i)*} y_j^{(i)} \exp\left(-\gamma_i \left\| \tilde{x}_j^{(i)} - \tilde{x}^{(i)} \right\|^2 \right) + b_i. \tag{27}$$

Here, γ_i is the effective numerical RBF coefficient determined during local training and retained at client i; it remains unchanged during validation and testing.

Algorithm 3: Local RBF-SVM model training at client i

Input: Local dataset $\mathcal{D}_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{n_i}$; globally selected SVM regularization parameter $C_{\text{svm}} > 0$; globally selected RBF kernel setting $\gamma_{\text{cfg}} \in \{\texttt{scale}\} \cup \mathbb{R}_{>0}$; support-vector tolerance $\epsilon > 0$

Output: Client-local model state $(\mathcal{M}_i, b_i, \gamma_i, \mu_i, \sigma_i)$

`/* - Local Preprocessing - */`

Compute μ_i and the feature-scaling vector σ_i exclusively from $\mathcal{D}_i$, assigning a scaling factor of 1 to each locally constant feature;

for $j = 1$ **to** n_i **do**

$$\tilde{x}_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_i}{\sigma_i};$$

Construct the standardized local training matrix

$$\tilde{X}_i \leftarrow \left[\tilde{x}_1^{(i)}, \ldots, \tilde{x}_{n_i}^{(i)}\right]^\top;$$

`/* - Effective RBF-Coefficient Computation - */`

if $\gamma_{\text{cfg}} =$ *scale* **then**

$$\gamma_i \leftarrow \frac{1}{d \operatorname{Var}(\tilde{X}_i)};$$

else

$$\gamma_i \leftarrow \gamma_{\text{cfg}};$$

`/* - Local Kernel-Matrix Construction - */`

Initialize $K^{(i)} \in \mathbb{R}^{n_i \times n_i}$;

for $j = 1$ **to** n_i **do**

  for $k = 1$ **to** n_i **do**

$$K_{jk}^{(i)} \leftarrow \exp\left(-\gamma_i \left\| \tilde{x}_j^{(i)} - \tilde{x}_k^{(i)} \right\|^2\right);$$

`/* - Local SVM Training - */`

Solve the local SVM dual problem in Equation (18), subject to Equations (19) and (20), to obtain $\alpha^{(i)*}$;

Identify the local support-vector set

$$\mathcal{S}_i \leftarrow \left\{ j \,\middle|\, \alpha_j^{(i)*} > \epsilon \right\};$$

Identify the local margin support-vector set

$$\mathcal{S}_i^\circ \leftarrow \left\{ j \,\middle|\, \epsilon < \alpha_j^{(i)*} < C_{\text{svm}} - \epsilon \right\};$$

if $\left|\mathcal{S}_i^\circ\right| > 0$ **then**

  Compute b_i using Equation (23);

else

  Retain the bias term returned by the underlying local SVM implementation;

`/* - Local Model Construction and Retention - */`

Construct the local support-vector model

$$\mathcal{M}_i \leftarrow \left\{ \left(\tilde{x}_j^{(i)}, y_j^{(i)}, \alpha_j^{(i)*}\right) \,\middle|\, j \in \mathcal{S}_i \right\};$$

Retain $\mathcal{M}_i$, b_i, γ_i, μ_i, and σ_i at client i;

return $(\mathcal{M}_i, b_i, \gamma_i, \mu_i, \sigma_i)$ *locally at client i; no model state is transmitted to the server*

After receiving one scalar score from each participating client, the server computes the federated decision score as

$$s_{\text{Fed}}(x) = \frac{1}{M} \sum_{i \in \mathcal{C}} s_i(x), \qquad M = |\mathcal{C}|, \tag{28}$$

where arithmetic averaging is adopted as the decision-score aggregation rule. The server therefore performs decision-level aggregation rather than model-parameter aggregation.

Although all clients use a common C_{svm}, a common kernel configuration γ_{cfg}, a common feature representation, and the same preprocessing procedure, independently trained RBF-SVMs are not formally guaranteed to produce decision-function scores on identical numerical scales. Moreover, when $\gamma_{\text{cfg}} = \texttt{scale}$, the effective coefficients γ_i may differ across clients because they are computed from client-specific standardized training matrices. Arithmetic averaging of the local decision scores therefore constitutes a design assumption of the proposed approach; future work may investigate score normalization, calibration, or weighted aggregation before server-side fusion.

The resulting federated decision score is converted into the final binary prediction using the calibrated decision threshold τ:

$$\hat{y} = \begin{cases} +1, & s_{\text{Fed}}(x) \geq \tau, \\ -1, & s_{\text{Fed}}(x) < \tau. \end{cases} \tag{29}$$

Algorithm 4 summarizes the complete federated-inference procedure, including server-side client selection and score aggregation as well as client-side decision-score computation.

Privacy and Deployment Considerations

The proposed architecture preserves the locality of historical training data and locally trained RBF-SVM models because neither is transmitted to the coordinating server. However, this locality should not be interpreted as a formal privacy guarantee.

During collaborative inference, the d-dimensional query vector must be made available to the participating clients. Consequently, the current formulation does not protect the confidentiality of the inference sample from the clients involved in the prediction. This limitation is particularly relevant when participating wells belong to different operators or administrative domains. The formulation therefore assumes a trusted or explicitly authorized collaborative environment in which sharing the inference sample is permitted.

A second privacy consideration arises from the transmission of exact real-valued decision-function scores. Although local training samples, support vectors, and model parameters are not transmitted, repeated access to the local decision function may expose information about the trained classifier and expose it to model-extraction or membership-inference attacks [44,45]. Possible mitigation mechanisms include score quantization, calibrated noise, inference-query restrictions, trusted execution environments, encrypted inference, and label-only aggregation. These mechanisms may introduce trade-offs among privacy, communication cost, score-aggregation fidelity, threshold calibration, and anomaly-detection performance, and evaluating these trade-offs is left for future work.

Unlike conventional federated learning methods based on model-parameter aggregation, such as FedAvg, FedRBF-SVM does not construct or iteratively update a shared global model. Instead, independently trained local RBF-SVMs remain at their respective clients, while the coordinating server receives and aggregates only their scalar decision-function scores during collaborative inference.

Algorithm 4: FedRBF-SVM federated inference

Input: Number of clients K; client participation fraction $C_f \in (0, 1]$; client-local model states $\{(\mathcal{M}_i, b_i, \gamma_i, \mu_i, \sigma_i)\}_{i=1}^{K}$, with each state retained at its corresponding client;
inference sample $x \in \mathbb{R}^d$; calibrated decision threshold τ
Output: Predicted class label $\hat{y} \in \{-1, +1\}$
`/* - Server Processing -` */
Select the participating-client subset $\mathcal{C} \subseteq \{1, \ldots, K\}$ such that $|\mathcal{C}| \leftarrow \max\left(\left\lfloor C_f K \right\rfloor, 1\right)$;
Set $M \leftarrow |\mathcal{C}|$;
Make the inference sample x available to each participating client $i \in \mathcal{C}$;
for *each client* $i \in \mathcal{C}$ *in parallel* **do**
  $s_i(x) \leftarrow \textsc{LocalScore}(i, x)$;
Compute the federated decision score
$s_{\text{Fed}}(x) \leftarrow \frac{1}{M} \sum_{i \in \mathcal{C}} s_i(x)$;
if $s_{\text{Fed}}(x) \geq \tau$ **then**
  $\hat{y} \leftarrow +1$;
else
  $\hat{y} \leftarrow -1$;
return $\hat{y}$
`/* - Client Processing -` */
Procedure *LocalScore*(i, x)**:**
  Standardize the inference sample using the locally retained preprocessing parameters
  $\tilde{x}^{(i)} \leftarrow \frac{x - \mu_i}{\sigma_i}$;
  Compute the local scalar decision-function score
  $s_i(x) \leftarrow \sum_{(\tilde{x}_j^{(i)}, y_j^{(i)}, \alpha_j^{(i)*}) \in \mathcal{M}_i} \alpha_j^{(i)*} y_j^{(i)} \exp\left(-\gamma_i \left\| \tilde{x}_j^{(i)} - \tilde{x}^{(i)} \right\|^2\right) + b_i$;
  return $s_i(x)$ *to the coordinating server*

4. FedRBF-SVM Evaluation

This section describes the framework used to evaluate FedRBF-SVM as a binary anomaly-detection approach. The anomaly class is assigned the positive label $(+1)$, and the normal class is assigned the negative label (-1). The evaluation focuses on precision, recall, the true-negative rate (TNR), the false-positive rate (FPR), the false-negative rate (FNR), the F_1-score, and the Matthews correlation coefficient (MCC). Collectively, these metrics characterize anomaly-detection sensitivity, false-alarm behavior, missed detections, alert reliability, specificity, and overall binary classification performance.

For each evaluated sample, the participating clients perform the local standardization and decision-score computations defined in Equations (26) and (27). The resulting local scores are aggregated according to Equation (28), and the calibrated threshold τ is then applied according to Equation (29).

For a conventional centralized RBF-SVM, the standard decision boundary corresponds to a decision-function threshold of $\tau = 0$. In FedRBF-SVM, however, the aggregated decision score is obtained by averaging the decision-function scores of independently trained local classifiers. Consequently, a zero threshold does not necessarily correspond to

the desired operational FPR. The FedRBF-SVM operating threshold is therefore calibrated using only validation data.

Let α denote the target FPR. The target operating points considered in this study are $\alpha \in \{0.01, 0.03, 0.05\}$. For each α, the corresponding decision threshold is obtained from the aggregated decision scores of the normal validation samples as

$$\tau_\alpha = Q_{1-\alpha}\big(\{s_{\mathrm{Fed}}(x_q) \,|\, y_q = -1,\ x_q \in \mathcal{D}_{\mathrm{val}}\}\big), \tag{30}$$

where $Q_{1-\alpha}(\cdot)$ denotes the $(1-\alpha)$ quantile of the empirical federated decision-score distribution for the normal validation samples. The calibrated threshold τ_α is then applied unchanged to the independent test set, ensuring that no test-set information is used during threshold selection. Validation samples are processed using the same client-participation and score-aggregation procedures as test samples, so threshold calibration and test evaluation use the same participation fraction and client-selection procedure.

Because precision, recall, TNR, FPR, FNR, F_1-score, and MCC depend on the selected operating threshold, comparisons between the FedRBF-SVM approach and a centralized reference classifier must account for differences in threshold selection. Evaluating a centralized SVM at $\tau = 0$ characterizes its performance at its default operating point, whereas direct comparisons at a prescribed FPR are most meaningful when equivalent validation-based threshold-calibration procedures are applied to both FedRBF-SVM and the centralized reference classifier.

4.1. Performance Metrics

Let the independent test set be denoted by

$$\mathcal{D}_{\mathrm{test}} = \{(x_q, y_q)\}_{q=1}^{n_{\mathrm{test}}}. \tag{31}$$

The confusion-matrix components are defined as follows:

$$TP = \sum_{q=1}^{n_{\mathrm{test}}} \mathbb{1}\big[y_q = +1 \wedge \hat{y}_q = +1\big], \tag{32}$$

$$TN = \sum_{q=1}^{n_{\mathrm{test}}} \mathbb{1}\big[y_q = -1 \wedge \hat{y}_q = -1\big], \tag{33}$$

$$FP = \sum_{q=1}^{n_{\mathrm{test}}} \mathbb{1}\big[y_q = -1 \wedge \hat{y}_q = +1\big], \tag{34}$$

$$FN = \sum_{q=1}^{n_{\mathrm{test}}} \mathbb{1}\big[y_q = +1 \wedge \hat{y}_q = -1\big], \tag{35}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function. TP denotes correctly detected anomalies, TN correctly identified normal samples, FP normal samples incorrectly classified as anomalies (false alarms), and FN anomalous samples incorrectly classified as normal (missed anomalies).

Accuracy, which is reported for the benchmark analysis in Section 5.4, measures the proportion of all test samples that are correctly classified and is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{36}$$

Precision measures the proportion of predicted anomalies that correspond to actual abnormal samples and is defined as

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{defined as 0 if } TP + FP = 0. \tag{37}$$

Precision is, however, sensitive to class prevalence and should therefore be interpreted in the context of the class distribution and alongside complementary measures such as FPR and MCC. This consideration is particularly important for the processed dataset used in this study because abnormal windows constitute the majority of the test set.

Recall, also referred to as sensitivity or the true-positive rate (TPR), measures the proportion of actual anomalies correctly detected and is defined as

$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN}, \qquad \text{defined as 0 if } TP + FN = 0. \tag{38}$$

Recall is particularly important in anomaly detection because false negatives correspond to undetected abnormal events that may lead to operational disruptions, safety risks, or production losses.

TNR, also referred to as specificity, measures the proportion of normal samples correctly identified and is defined as

$$\text{TNR} = \frac{TN}{TN + FP}, \qquad \text{defined as 0 if } TN + FP = 0. \tag{39}$$

FPR measures the proportion of normal samples incorrectly classified as anomalies and is defined as

$$\text{FPR} = \begin{cases} \dfrac{FP}{FP + TN}, & FP + TN > 0, \\ 0, & FP + TN = 0. \end{cases} \tag{40}$$

When $FP + TN > 0$, FPR is the complement of TNR, such that FPR $= 1 -$ TNR.

A lower FPR indicates fewer false alarms. In oil and gas well monitoring, controlling the FPR is important because excessive false alarms can reduce operator confidence in the monitoring system and lead to unnecessary operational interventions.

FNR measures the proportion of anomalous samples that remain undetected and is defined as

$$\text{FNR} = \begin{cases} \dfrac{FN}{FN + TP}, & FN + TP > 0, \\ 0, & FN + TP = 0. \end{cases} \tag{41}$$

When $FN + TP > 0$, FNR is the complement of TPR, such that FNR $= 1 -$ TPR.

A lower FNR indicates that fewer abnormal events are missed. FNR and FPR should therefore be considered jointly, particularly in safety-sensitive anomaly-detection applications, where reducing missed events must be balanced against maintaining an acceptable false-alarm rate.

The F_1-score combines precision and recall through their harmonic mean. Using the confusion-matrix components, it is defined as

$$F_1 = \begin{cases} \dfrac{2TP}{2TP + FP + FN}, & 2TP + FP + FN > 0, \\ 0, & 2TP + FP + FN = 0. \end{cases} \tag{42}$$

When Precision + Recall > 0, this definition is equivalent to

$$F_1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{43}$$

The F_1-score can be useful for anomaly-detection problems involving class imbalance because it jointly considers precision and recall. However, because it does not include true negatives, the F_1-score should be interpreted alongside FPR and MCC.

MCC incorporates all four confusion-matrix components and is defined as

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{44}$$

In this study, MCC is set to 0 when its denominator is zero. MCC ranges from -1 to $+1$:$+1$ indicates perfect agreement between the predicted and true labels, 0 no overall correlation, and -1 perfect disagreement. Because it uses all four components, MCC provides a complementary assessment of classification performance when class proportions are unequal, thereby reducing reliance on positive-class measures such as precision or recall. This is particularly relevant to the processed dataset considered here, in which abnormal windows constitute the majority class.

4.2. Communication and Computational Costs

The communication cost of FedRBF-SVM comprises two components: distributing an inference sample to the participating clients and transmitting the resulting scalar decision scores to the server. Because the mechanism used to distribute the inference sample depends on the deployment architecture, these two components are considered separately.

Let each decision-function score be encoded using b_s bits. For each inference query, every participating client among the $M = |\mathcal{C}|$ clients returns one scalar score to the server. Therefore, the client-to-server communication cost is

$$C_{\text{score}} = Mb_s \quad \text{bits}, \tag{45}$$

corresponding to a communication complexity of $O(M)$.

If the coordinating server must explicitly transmit the d-dimensional inference vector to each of the M participating clients, and each feature value is encoded using b_x bits, the point-to-point query-distribution cost is

$$C_{\text{query}} = Mdb_x \quad \text{bits}. \tag{46}$$

If the inference sample originates from one of the participating clients, only the remaining $M - 1$ clients need to receive the query, reducing the query-distribution cost to $(M - 1)db_x$ bits. Broadcast or multicast communication may further reduce the effective query-distribution overhead.

Under the point-to-point server-to-client distribution model, the total communication cost per inference query is

$$C_{\text{total}} = Mdb_x + Mb_s = M(db_x + b_s) \quad \text{bits}. \tag{47}$$

For comparison, under the same communication-accounting convention, a conventional centralized inference architecture requires the d-dimensional query to be transmitted once to the central server and one scalar decision score to be returned. The resulting communication cost is approximately $db_x + b_s$ bits per query when protocol overhead is neglected. Under the point-to-point FedRBF-SVM architecture defined in Equation (47), the corresponding cost

is $M(db_x + b_s)$ bits.Therefore, the point-to-point communication cost of FedRBF-SVM is M times the centralized per-query cost, or five times for the $M = 5$ clients used in the experiments.

Consequently, FedRBF-SVM should not be interpreted as providing a lower end-to-end communication cost under point-to-point query distribution. Its communication efficiency applies specifically to the compact client-to-server response because each participating client returns only one scalar score; no local training data, support vectors, or model parameters are transmitted. If the inference sample is already available at the participating clients or broadcast or multicast is used, the query-distribution component can be reduced accordingly.

The server-side computational cost of score aggregation is linear in the number of participating clients: computing $s_{\mathrm{Fed}}(x)$ in Equation (28) requires $M - 1$ additions and one division, resulting in an aggregation complexity of $O(M)$.

Local model training also incurs computational and memory costs at each client. For client i, constructing the complete RBF kernel matrix requires evaluating the kernel for all pairs of n_i local training samples. Because each kernel evaluation operates on d features, explicit kernel-matrix construction has a computational complexity of $O(n_i^2 d)$ and a memory complexity of $O(n_i^2)$ when the complete matrix is retained. The subsequent cost of solving the SVM dual optimization problem depends on the numerical solver and the characteristics of the local dataset. These training computations are performed locally and do not require the transmission of the kernel matrix, support vectors, or local training samples to the server.

The main inference computation is performed by the participating clients. For client i, evaluating the RBF-SVM decision function requires computing the RBF kernel between the standardized query vector and each locally retained support vector. Standardizing the query has a computational complexity of $O(d)$, whereas computing the RBF-SVM score has a complexity of $O(|\mathcal{S}_i|d)$. Therefore, the overall local inference complexity is $O(|\mathcal{S}_i|d)$ per query and can be reduced further using fast-prediction techniques for RBF-kernel SVMs [46]. The locally retained support vectors and model parameters are not transmitted to the server. Consequently, the communication cost of returning the score to the server is independent of both $|\mathcal{S}_i|$ and d.

The following section presents simulation results that quantify the performance of FedRBF-SVM for different numbers of locally stored training patterns and target FPR operating points and compares it with a single local RBF-SVM and the centralized RBF-SVM reference.

5. Results

This section describes the dataset and experimental setup, and presents the performance results.

5.1. Dataset and Preprocessing

The simulation framework was implemented in Python (version 3.13.3) [47] as an extension of version 1.0.0 of the 3W Toolkit [48] and used version 2.0.0 of the 3W dataset [22]. Released by Petrobras, the 3W Toolkit is an open-source Python package that supports experimentation with the 3W dataset by providing utilities for data loading, preprocessing, and the evaluation of undesirable-event detection methods in oil wells, thereby facilitating standardized data handling and reproducible experimentation [48]. The 3W dataset contains labeled multivariate time-series instances from three sources: real operating data, simulated data, and hand-drawn data generated by experts [22]. These instances represent normal operating conditions and multiple undesirable-event conditions, which makes the

dataset particularly relevant for investigating undesirable-event and anomaly-detection methods in offshore oil wells.

All base classes in the 3W dataset were retained, with class 0 representing normal operation and classes 1–9 representing undesirable-event conditions. Transient subclasses were mapped to their corresponding base classes. For the binary classification task, class 0 was assigned to the normal class (label -1), and all nonzero classes were assigned to the abnormal class (label $+1$).

Each multivariate time-series instance was transformed into fixed-length feature vectors by windowing. Each processed window contained 300 time steps from three retained sensor channels: `P-PDG` (pressure at the permanent downhole gauge), `P-TPT` (pressure at the temperature and pressure transducer), and `T-TPT` (temperature at the temperature and pressure transducer). The `class` column was used only as the target label and was excluded from the input feature vector. The three channels were flattened to form a feature vector of dimensionality $d = 300 \times 3 = 900$. The preprocessing configuration retained every fifth observation and used a window stride of 150 samples for windows of length 300, corresponding to 50% overlap between consecutive windows.

The processed dataset contained 70,361 windows in the original training partition and 14,731 windows in the test partition. The training partition comprised 13,114 normal and 57,247 abnormal windows (approximately 18.6% and 81.4%, respectively), and the test partition comprised 3593 normal and 11,138 abnormal windows (approximately 24.4% and 75.6%, respectively).

The original training partition was further divided by stratified random sampling into a training subset (80%) for model fitting and a validation subset (20%) for hyperparameter selection and decision-threshold calibration, yielding 56,288 training windows and 14,073 validation windows.

The rarity of undesirable events in the original 3W formulation should be distinguished from the class prevalence of the processed window-level dataset. The classification samples are fixed-length windows rather than independent undesirable-event occurrences; consequently, the window-level class proportions do not directly represent how often undesirable events occur in operation. Because all undesirable-event classes are merged into a single abnormal class, the processed dataset is skewed toward the abnormal class rather than representing a rare-positive-class setting, and the reported performance metrics should be interpreted in this context.

5.2. Experimental Setup

The FedRBF-SVM approach was evaluated in a simulated distributed setting with $K = 5$ clients, all of which participated in each evaluated prediction ($C_f = 1$ and $M = |\mathcal{C}| = 5$). These clients represent conceptual well-side processing nodes; the experiment did not partition the 3W data by individual physical wells. For each evaluated number of locally stored training patterns $k \in \{50, 100, 150, \ldots, 2500\}$, each client independently drew k patterns from the training subset. To ensure that a valid binary SVM could be trained locally, sampling was repeated when necessary until both classes were represented in each client's local dataset. The local models were thus trained on independent random samples drawn from the training subset, emulating the distributed availability of limited local training data. Because the client samples were drawn independently, they could overlap.

The centralized RBF-SVM reference was trained using k patterns, matching the number available at an individual client. In contrast, FedRBF-SVM trained M local models using a total of $M \times k$ client-level training observations; when client subsets overlapped, the number of unique observations available collectively could be smaller than $M \times k$.

For the principal comparative evaluation, a grid search was applied to both the centralized RBF-SVM and FedRBF-SVM, using $C_{\text{svm}} \in \{0.1, 1, 10, 100\}$ and $\gamma_{\text{cfg}} \in \{\texttt{scale}, 0.01, 0.1, 1.0\}$. For the centralized RBF-SVM, the pair $(C_{\text{svm}}, \gamma_{\text{cfg}})$ was selected separately for each value of k as the configuration that maximized the anomaly-class F_1-score on the validation subset, with MCC as the tie-breaker. The centralized reference retained its conventional decision threshold of $\tau = 0$.

For FedRBF-SVM, hyperparameter selection was performed separately for each combination of k and target FPR $\alpha \in \{0.01, 0.03, 0.05\}$. For each candidate $(C_{\text{svm}}, \gamma_{\text{cfg}})$ pair, the threshold τ_α was first calibrated on the validation subset using Equation (30), and the candidate that maximized the anomaly-class F_1-score at the resulting operating point was selected, with MCC as the tie-breaker. The calibrated threshold was then fixed and applied unchanged to the test partition. All RBF-SVM models were trained using the `SVC` implementation of scikit-learn [42]; therefore, no separate user-defined tolerance ϵ was applied to the dual coefficients for support-vector identification. The entire procedure was repeated for 100 independent repetitions, with hyperparameters reselected in each repetition; consequently, the selected $(C_{\text{svm}}, \gamma_{\text{cfg}})$ pair could vary across repetitions for a given combination of k and α. A centrally held validation subset was used to provide a controlled and reproducible comparison.

In an operational deployment, hyperparameter selection and threshold calibration need not rely on a centrally held labeled validation set. Each participating client can retain a labeled local validation subset and use it to evaluate candidate $(C_{\text{svm}}, \gamma_{\text{cfg}})$ configurations through the same collaborative decision-score procedure used during inference, in which each validation sample is scored by all participating local models before aggregation.

For threshold calibration at a target FPR α, the coordinating server can conduct a distributed threshold search. For each candidate threshold, the server returns the aggregated score of each validation sample to the client that owns it; that client counts its normal samples whose aggregated scores exceed the threshold and returns only this count. The server combines these counts to identify the threshold τ_α that most closely achieves the prescribed operating point. After threshold calibration, each client evaluates the corresponding $(C_{\text{svm}}, \gamma_{\text{cfg}})$ configuration using its local validation subset and returns only aggregate confusion-matrix statistics. The server then combines these statistics to compute the global anomaly-class F_1-score and MCC, allowing the same hyperparameter-selection criterion used in the present experiments to be applied without transmitting raw validation labels or pooling the validation subsets.

This distributed procedure remains subject to the query-sharing assumption of FedRBF-SVM because each validation feature vector must be made available to the clients participating in its collaborative evaluation. Thus, the centrally held validation subset used in this study is an experimental convenience rather than a requirement of the proposed approach. The principal comparison is subject to the experimental-design limitations summarized in Section 6.

5.3. Principal Performance Evaluation

Table 1 summarizes the mean $\pm$ standard deviation of the principal performance metrics over 100 independent repetitions. The centralized RBF-SVM is reported once for each value of k because it uses the conventional threshold $\tau = 0$, whereas FedRBF-SVM is reported at the three target FPR operating points. These results, examined in detail in the following subsections, illustrate the trade-off between false-alarm control and anomaly-detection performance as the number of locally stored training patterns per client increases.

Table 1. Mean ± standard deviation of selected anomaly-detection performance metrics over 100 independent repetitions for the centralized RBF-SVM and FedRBF-SVM. For FedRBF-SVM, k denotes the number of locally stored training patterns per client, whereas for the centralized RBF-SVM, k denotes the total number of training patterns. The centralized RBF-SVM uses the conventional threshold $\tau = 0$.

Training Patterns, k	Approach	Target FPR, α	F_1-Score	Precision	Recall	TNR	FPR	MCC
50	Centralized RBF-SVM	–	0.675 ± 0.108	0.784 ± 0.029	0.608 ± 0.161	0.478 ± 0.145	0.522 ± 0.145	0.169
	FedRBF-SVM	0.01	0.242 ± 0.038	0.977 ± 0.004	0.139 ± 0.024	0.990 ± 0.001	0.010 ± 0.001	0.256
	FedRBF-SVM	0.03	0.352 ± 0.080	0.960 ± 0.013	0.218 ± 0.059	0.970 ± 0.014	0.030 ± 0.014	0.310
	FedRBF-SVM	0.05	0.423 ± 0.096	0.942 ± 0.024	0.278 ± 0.077	0.942 ± 0.031	0.058 ± 0.031	0.334
500	Centralized RBF-SVM	–	0.621 ± 0.157	0.836 ± 0.038	0.526 ± 0.212	0.655 ± 0.224	0.345 ± 0.224	0.274
	FedRBF-SVM	0.01	0.400 ± 0.019	0.989 ± 0.001	0.251 ± 0.015	0.991 ± 0.001	0.009 ± 0.001	0.368
	FedRBF-SVM	0.03	0.549 ± 0.037	0.972 ± 0.004	0.383 ± 0.036	0.965 ± 0.007	0.035 ± 0.007	0.451
	FedRBF-SVM	0.05	0.669 ± 0.006	0.956 ± 0.002	0.514 ± 0.007	0.927 ± 0.003	0.073 ± 0.003	0.520
1000	Centralized RBF-SVM	–	0.680 ± 0.186	0.862 ± 0.030	0.600 ± 0.227	0.679 ± 0.167	0.321 ± 0.167	0.369
	FedRBF-SVM	0.01	0.496 ± 0.018	0.990 ± 0.005	0.331 ± 0.017	0.989 ± 0.005	0.011 ± 0.005	0.434
	FedRBF-SVM	0.03	0.606 ± 0.019	0.974 ± 0.003	0.440 ± 0.020	0.963 ± 0.006	0.037 ± 0.006	0.494
	FedRBF-SVM	0.05	0.695 ± 0.019	0.960 ± 0.002	0.545 ± 0.022	0.929 ± 0.004	0.071 ± 0.004	0.547
1500	Centralized RBF-SVM	–	0.706 ± 0.123	0.864 ± 0.003	0.613 ± 0.161	0.700 ± 0.080	0.300 ± 0.080	0.398
	FedRBF-SVM	0.01	0.557 ± 0.015	0.991 ± 0.002	0.388 ± 0.015	0.989 ± 0.002	0.011 ± 0.002	0.479
	FedRBF-SVM	0.03	0.657 ± 0.007	0.976 ± 0.001	0.495 ± 0.008	0.962 ± 0.003	0.038 ± 0.003	0.538
	FedRBF-SVM	0.05	0.720 ± 0.017	0.962 ± 0.003	0.575 ± 0.022	0.930 ± 0.005	0.070 ± 0.005	0.573
2000	Centralized RBF-SVM	–	0.733 ± 0.087	0.867 ± 0.007	0.644 ± 0.123	0.695 ± 0.056	0.305 ± 0.056	0.423
	FedRBF-SVM	0.01	0.614 ± 0.017	0.993 ± 0.000	0.444 ± 0.018	0.991 ± 0.000	0.009 ± 0.000	0.525
	FedRBF-SVM	0.03	0.700 ± 0.010	0.979 ± 0.001	0.545 ± 0.012	0.963 ± 0.003	0.037 ± 0.003	0.578
	FedRBF-SVM	0.05	0.758 ± 0.009	0.967 ± 0.003	0.624 ± 0.012	0.934 ± 0.006	0.066 ± 0.006	0.615
2500	Centralized RBF-SVM	–	0.738 ± 0.077	0.871 ± 0.005	0.648 ± 0.114	0.704 ± 0.050	0.296 ± 0.050	0.434
	FedRBF-SVM	0.01	0.667 ± 0.007	0.993 ± 0.001	0.502 ± 0.008	0.989 ± 0.001	0.011 ± 0.001	0.569
	FedRBF-SVM	0.03	0.740 ± 0.003	0.979 ± 0.001	0.594 ± 0.004	0.961 ± 0.003	0.039 ± 0.003	0.615
	FedRBF-SVM	0.05	0.790 ± 0.009	0.969 ± 0.001	0.667 ± 0.014	0.934 ± 0.003	0.066 ± 0.003	0.652

5.4. Analysis of Decision-Score Aggregation

To separate the contribution of decision-score aggregation from the effects of local training and aggregate data availability, three configurations were compared at the same validation-calibrated target FPR. The first, denoted Single, is a single RBF-SVM trained using $k = 1000$ training patterns and represents anomaly classification without collaborative aggregation. The second is FedRBF-SVM with five participating clients, each of which independently trains a local RBF-SVM using $k = 1000$ training patterns; during inference, the five local decision-function scores are averaged to obtain the federated decision score. The third, denoted Centralized, is a centralized RBF-SVM trained using $5k = 5000$ training patterns, which matches the nominal aggregate number of client-level training observations across the five clients.

The comparison between Single and FedRBF-SVM assesses the contribution of collaborative decision-score aggregation relative to a single locally trained classifier, whereas the comparison between FedRBF-SVM and Centralized indicates the performance attainable when the nominal aggregate training data are pooled for centralized training. All three configurations were evaluated at a target FPR of $\alpha = 0.05$, with thresholds calibrated using the same validation-based quantile procedure (Equation (30)).

Unlike the principal evaluation (Section 5.3), in which $(C_{\text{svm}}, \gamma_{\text{cfg}})$ was selected by grid search for each configuration, this analysis fixed the hyperparameters at $C_{\text{svm}} = 1$ and $\gamma_{\text{cfg}} =$ `scale` for all three configurations; under the `scale` setting, each model computed its effective RBF coefficient from its own standardized training matrix. This fixed-hyperparameter design isolates the effect of collaborative decision-score aggregation and avoids confounding the comparison with differences in hyperparameter selection. Consequently, the values in Table 2 are not expected to match the grid-search-optimized values in Table 1 and should not be interpreted as the best achievable performance of each method.

Table 2. Mean ± standard deviation over 100 independent repetitions for the decision-score aggregation analysis at a target FPR of $\alpha = 0.05$. The Single model and each FedRBF-SVM client use $k = 1000$ training patterns, whereas the Centralized model uses $5k = 5000$ training patterns. All configurations use $C_{\text{svm}} = 1$ and $\gamma_{\text{cfg}} =$ `scale`.

Configuration	Precision	Recall	F_1-Score	FPR	MCC	Accuracy
Single	0.909 ± 0.016	0.249 ± 0.069	0.387 ± 0.086	0.074 ± 0.009	0.184 ± 0.048	0.414 ± 0.050
FedRBF-SVM	0.924 ± 0.008	0.289 ± 0.053	0.438 ± 0.064	0.072 ± 0.007	0.218 ± 0.034	0.445 ± 0.038
Centralized	0.928 ± 0.005	0.349 ± 0.026	0.507 ± 0.028	0.083 ± 0.006	0.253 ± 0.018	0.487 ± 0.019

Table 2 shows that FedRBF-SVM achieved higher mean precision, recall, F_1-score, MCC, and accuracy than Single while maintaining a similar mean FPR. The mean F_1-score increased from 0.387 ± 0.086 to 0.438 ± 0.064, recall from 0.249 ± 0.069 to 0.289 ± 0.053, MCC from 0.184 ± 0.048 to 0.218 ± 0.034, accuracy from 0.414 ± 0.050 to 0.445 ± 0.038, and precision from 0.909 ± 0.016 to 0.924 ± 0.008. The mean realized FPR changed only slightly, from 0.074 ± 0.009 to 0.072 ± 0.007. These results suggest that collaborative decision-score aggregation is associated with improved performance relative to isolated local inference under the evaluated conditions.

The Centralized configuration achieved the highest mean F_1-score, recall, MCC, and accuracy (0.507 ± 0.028, 0.349 ± 0.026, 0.253 ± 0.018, and 0.487 ± 0.019, respectively), compared with 0.438 ± 0.064, 0.289 ± 0.053, 0.218 ± 0.034, and 0.445 ± 0.038 for FedRBF-SVM. Mean precision was similar for the two configurations, whereas the Centralized model had a higher mean realized FPR of 0.083 ± 0.006, compared with 0.072 ± 0.007 for FedRBF-SVM.

Overall, the experiment yielded two complementary findings. First, the differences between Single and FedRBF-SVM suggest that collaborative decision-score aggregation can improve anomaly-classification performance relative to isolated local inference. Second, the higher mean F_1-score, recall, MCC, and accuracy of Centralized indicate that jointly training a single RBF-SVM on the nominal aggregate training data remains advantageous when data centralization is feasible. FedRBF-SVM should therefore be interpreted as a distributed alternative for collaborative inference that retains training data and trained models at their respective clients, rather than as an approach intended to outperform unconstrained centralized training.

5.5. F_1-Score

Figure 1 presents the F_1-scores of FedRBF-SVM at the three target FPR operating points and of the centralized RBF-SVM reference. FedRBF-SVM performance depended clearly on k: across all three target FPRs, the F_1-score generally increased with k. At the smallest evaluated value ($k = 50$), the FedRBF-SVM configurations had relatively low F_1-scores (0.24–0.42), reflecting their low recall under the imposed FPR constraints. By $k = 2500$, the F_1-score had risen to 0.67–0.79, indicating that larger local training datasets are associated with more effective decision-score aggregation.

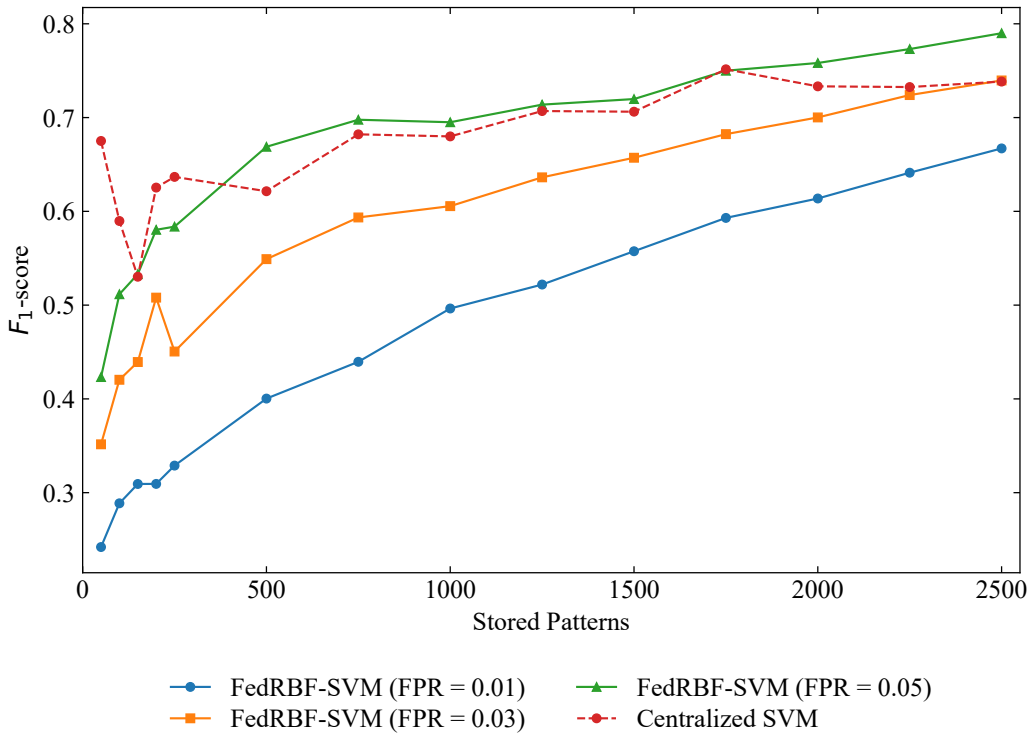


Figure 1. F_1-score as a function of the number of locally stored training patterns (k) for FedRBF-SVM at target FPR operating points $\alpha \in \{0.01, 0.03, 0.05\}$, compared with the centralized RBF-SVM reference.

Among the FedRBF-SVM operating points, $\alpha = 0.05$ generally achieved the highest F_1-score, followed by $\alpha = 0.03$ and $\alpha = 0.01$, reflecting the expected trade-off between stricter false-alarm control and anomaly-detection sensitivity. The comparison with the centralized reference is subject to the experimental-design limitations summarized in Section 6.

5.6. Recall

Figure 2 presents anomaly-class recall, which quantifies anomaly-detection sensitivity, as a function of k for FedRBF-SVM and the centralized RBF-SVM reference. The centralized

reference generally had higher recall than FedRBF-SVM at the stricter target FPRs. However, this difference should be interpreted together with the FPR results in Figure 3: because the centralized model used its conventional threshold of $\tau = 0$, it operated at substantially higher realized FPRs of approximately 0.30–0.52 across the evaluated values of k (Section 6).

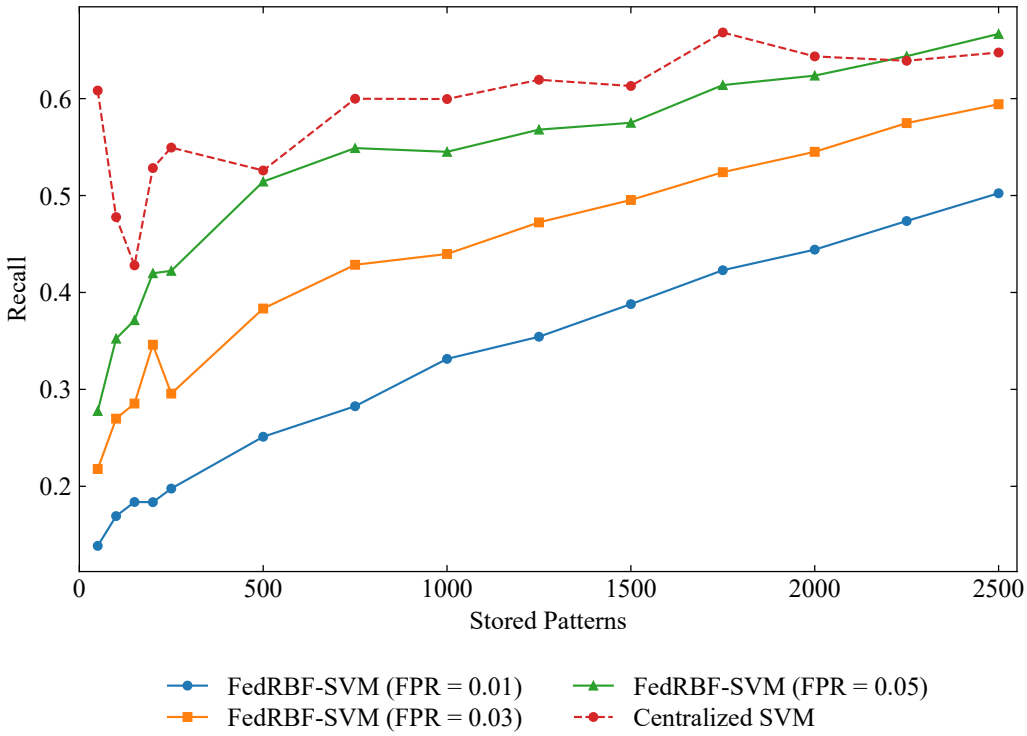


Figure 2. Recall as a function of the number of locally stored training patterns (k) for FedRBF-SVM at target FPR operating points $\alpha \in \{0.01, 0.03, 0.05\}$, compared with the centralized RBF-SVM reference.

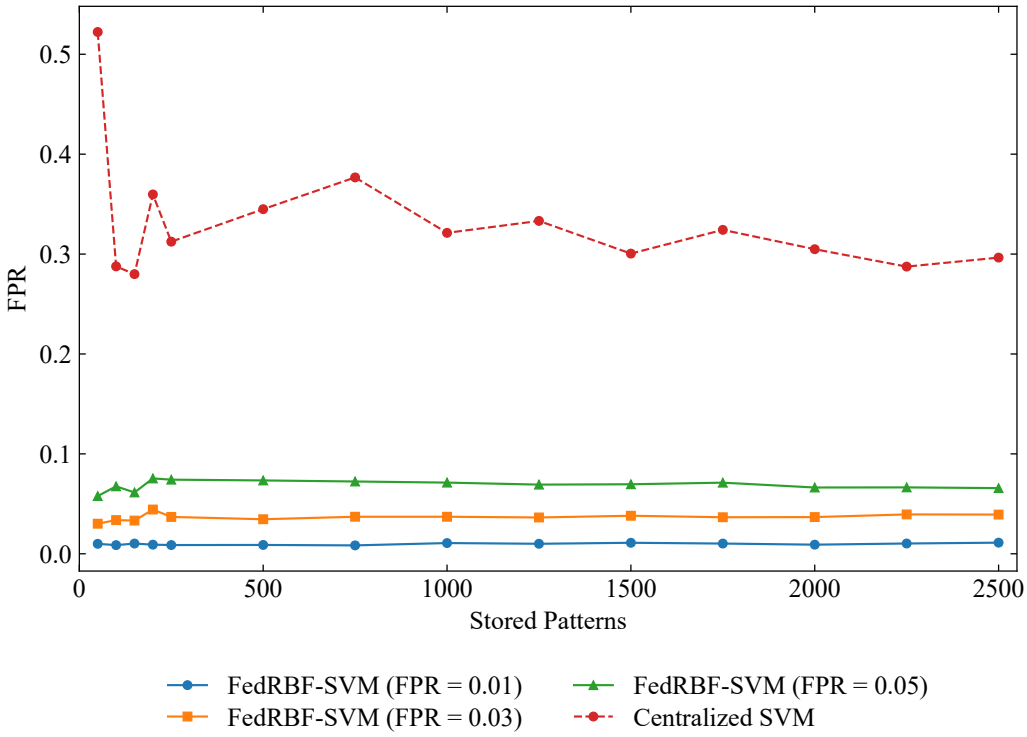


Figure 3. FPR as a function of the number of locally stored training patterns (k) for FedRBF-SVM at target FPR operating points $\alpha \in \{0.01, 0.03, 0.05\}$, compared with the centralized RBF-SVM reference.

For FedRBF-SVM, recall generally increased with k across all three target FPRs. At $k = 50$, recall ranged from approximately 0.14 for $\alpha = 0.01$ to 0.28 for $\alpha = 0.05$; by $k = 2500$, the $\alpha = 0.05$, 0.03, and 0.01 configurations achieved recall of approximately 0.67, 0.59, and 0.50, respectively. This ordering reflects the expected sensitivity trade-off associated with increasingly stringent false-alarm constraints.

Overall, increasing k was associated with a substantial improvement in the anomaly-detection sensitivity of FedRBF-SVM. The results also show a tunable trade-off: less restrictive target FPRs provided higher sensitivity, whereas stricter targets reduced false alarms at the expense of recall, allowing an operating point to be selected according to the relative importance of missed anomalies and false alarms.

5.7. False-Positive Rate

Figure 3 shows the realized FPR for the evaluated models. The centralized RBF-SVM, at its conventional threshold of $\tau = 0$, had FPRs from approximately 0.30 to 0.52. In contrast, FedRBF-SVM operated at substantially lower realized FPRs at the three validation-calibrated target operating points: approximately 0.009–0.011 for $\alpha = 0.01$, 0.030–0.039 for $\alpha = 0.03$, and 0.058–0.073 for $\alpha = 0.05$, the last representing an absolute deviation of 0.008–0.023 from the nominal target.

This difference must, however, be interpreted cautiously because the two approaches used different threshold-selection procedures: the centralized RBF-SVM used $\tau = 0$, whereas the FedRBF-SVM thresholds were calibrated on the validation subset to target specific FPRs. The results therefore demonstrate that FedRBF-SVM supports validation-based operating-point calibration for managing false-alarm behavior, rather than that it inherently produces fewer false alarms.

5.8. False-Negative Rate

Figure 4 presents the anomaly-class FNR as a function of k. Because the test partition contains anomalous samples, FNR equals $1 - \text{Recall}$; a lower FNR indicates fewer missed anomalies.

FedRBF-SVM showed a clear reduction in FNR across all three target FPRs as k increased. The $\alpha = 0.05$ configuration generally achieved the lowest FNR, followed by $\alpha = 0.03$ and $\alpha = 0.01$, consistent with the higher recall permitted by less restrictive FPR constraints.

These results reinforce the recall analysis: increasing k reduced the proportion of missed anomalies while FedRBF-SVM operated at validation-calibrated target FPRs. At larger values of k, the FNR of the $\alpha = 0.05$ configuration approached that of the centralized reference and fell slightly below it at $k = 2500$ (0.333 compared with 0.352). Because the centralized reference operated at a much higher FPR, this comparison is subject to the operating-point limitation discussed in Section 6.

5.9. Precision

Figure 5 presents anomaly-class precision. At $k = 50$, the three FedRBF-SVM configurations achieved precision of approximately 0.98 for $\alpha = 0.01$, 0.96 for $\alpha = 0.03$, and 0.94 for $\alpha = 0.05$; at $k = 2500$, the corresponding values were approximately 0.99, 0.98, and 0.97, respectively.

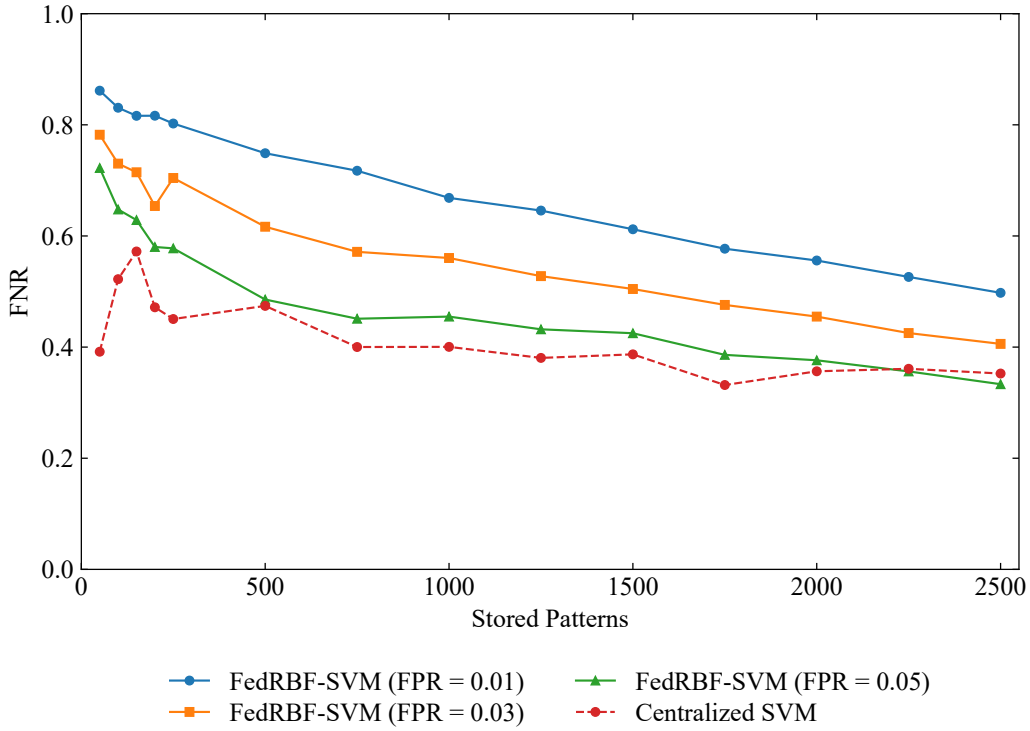


Figure 4. FNR as a function of the number of locally stored training patterns (k) for FedRBF-SVM at target FPR operating points $\alpha \in \{0.01, 0.03, 0.05\}$, compared with the centralized RBF-SVM reference.

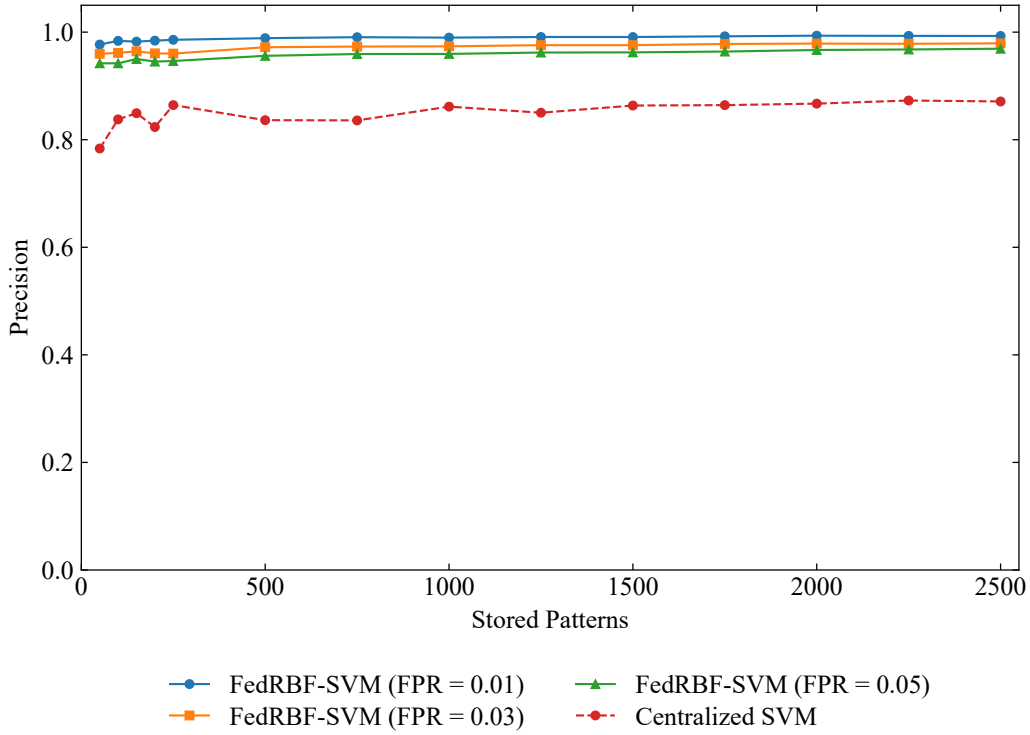


Figure 5. Precision as a function of the number of locally stored training patterns (k) for FedRBF-SVM at target FPR operating points $\alpha \in \{0.01, 0.03, 0.05\}$, compared with the centralized RBF-SVM reference.

These values should be interpreted in the context of the test-set class distribution, in which approximately 75.6% of windows are abnormal (Section 5.1). An always-abnormal classifier would therefore achieve a precision of approximately 0.756 solely because of the positive-class prevalence, although it would also have an FPR of 1.0, a specificity of

0, and an MCC of 0. The FedRBF-SVM precision values exceed this prevalence-based reference, but precision alone should not be interpreted as evidence of overall anomaly-detection effectiveness.

The centralized reference had a mean precision of approximately 0.78 at $k = 50$, increasing to approximately 0.83–0.87 for larger values of k. Although these values were lower than those of FedRBF-SVM, the comparison should be interpreted in light of differences in threshold selection and aggregate training-data availability, as well as the high abnormal-window prevalence of the test partition. Accordingly, precision should be interpreted alongside recall, FPR, FNR, and MCC. For example, at $k = 2500$, the $\alpha = 0.05$ configuration achieved a precision of approximately 0.97 and a recall of approximately 0.67, with a realized FPR of 0.066 and an MCC of 0.65.

5.10. Matthews Correlation Coefficient

Figure 6 reports MCC, which complements prevalence-sensitive measures such as precision on the abnormal-skewed test partition. The MCC of the centralized reference varied substantially at smaller values of k and then improved more gradually, reaching approximately 0.43 at $k = 2500$.

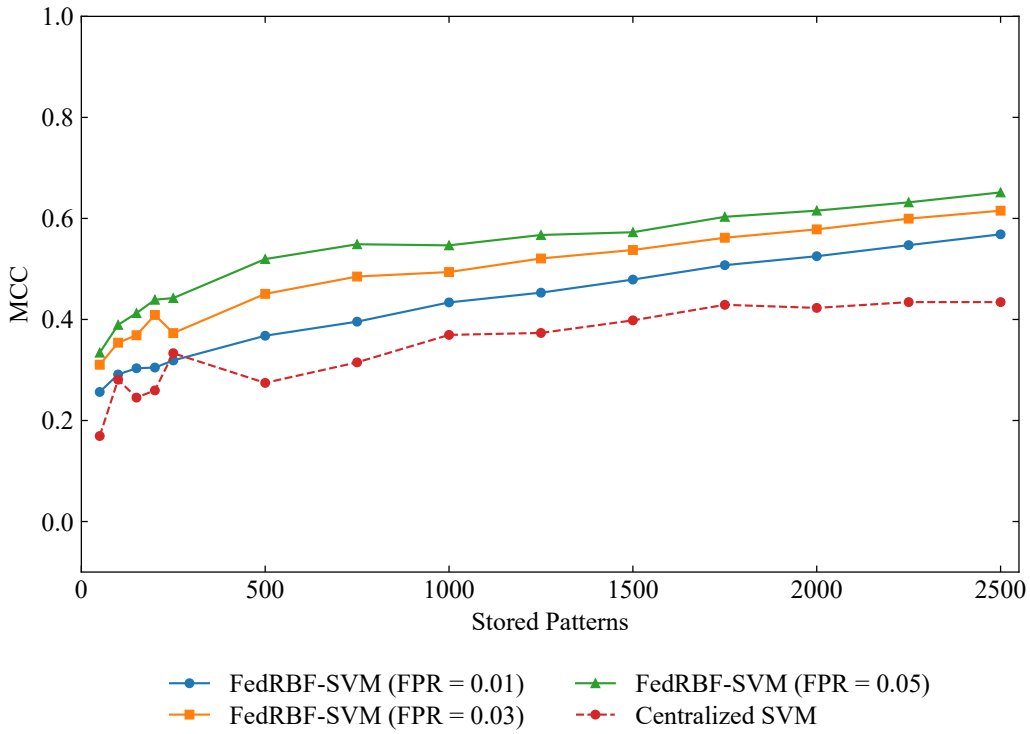


Figure 6. MCC as a function of the number of locally stored training patterns (k) for FedRBF-SVM at target FPR operating points $\alpha \in \{0.01, 0.03, 0.05\}$, compared with the centralized RBF-SVM reference.

Across all target FPRs, the MCC of FedRBF-SVM generally increased with k. At $k = 2500$, MCC reached approximately 0.57, 0.62, and 0.65 for $\alpha = 0.01$, 0.03, and 0.05, respectively, indicating improved overall classification balance as more local training data became available. The MCC comparison is subject to the experimental-design limitations summarized in Section 6.

6. Discussion

The experimental results highlight the effects of local training-data availability, threshold calibration, and collaborative decision-score aggregation on FedRBF-SVM performance. When only a small number of training patterns were available to each client, FedRBF-SVM

exhibited lower F_1-scores and recall than the centralized RBF-SVM reference, particularly under the stricter target FPRs. This behavior suggests that limited local training data can constrain the ability of independently trained RBF-SVM models to construct sufficiently representative decision boundaries. As the number of locally available training patterns increased at each client, this performance gap generally decreased. At larger values of k, the MCC of FedRBF-SVM generally increased, indicating improved overall classification performance across the evaluated configurations.

In upstream well monitoring, a detection method that achieves high anomaly sensitivity at the cost of excessive false alarms may have limited practical utility because frequent erroneous alerts can increase operator workload and lead to unnecessary inspections or interventions. The FedRBF-SVM approach maintained realized test-set FPRs close to the specified targets across the evaluated values of k. The $\alpha = 0.01$ and $\alpha = 0.03$ configurations remained particularly close to their respective target operating points, whereas larger deviations were observed for $\alpha = 0.05$. Because the decision thresholds were calibrated on the validation subset and applied unchanged to the test partition, such deviations are expected when the score distributions of the normal validation and test windows differ.

These results indicate that validation-based threshold calibration provides a practical mechanism for selecting a desired false-alarm operating region. In the benchmark analysis (Section 5.4), FedRBF-SVM outperformed a single local RBF-SVM at a similar mean FPR, whereas a centralized RBF-SVM trained on the pooled $5k$ training patterns remained superior, indicating that decision-score aggregation improves on isolated local inference but does not replace centralized joint training when pooling is feasible.

Taken together, recall, FNR, precision, and FPR illustrate the trade-off controlled by the target FPR in FedRBF-SVM. Less restrictive targets generally increased recall, and thus reduced FNR, whereas stricter targets suppressed false alarms at the expense of anomaly-detection sensitivity. Precision remained high (0.94–0.99) across the evaluated values of k, although these values should be read against the prevalence-based reference of 0.756 on the abnormal-skewed test partition (Section 5.9). For example, at $k = 2500$, the $\alpha = 0.05$ configuration achieved a recall of approximately 0.67 and a precision of approximately 0.97 at a realized FPR of 0.066. The target FPR can therefore be used to select a trade-off between missed anomalies and false alarms that suits the operational requirements of the monitoring application.

The increase in MCC with k indicates that the improvements were not limited to positive-class metrics such as recall and precision but extended to overall agreement between predicted and true labels across all four confusion-matrix components. At every value of k reported in Table 1, all three FedRBF-SVM configurations also achieved higher MCC than the centralized reference, from 0.26–0.33 compared with 0.17 at $k = 50$ to 0.57–0.65 compared with 0.43 at $k = 2500$. This comparison, however, remains subject to differences in aggregate training-data availability and threshold selection.

The analysis shows that the communication advantage of FedRBF-SVM lies in the compact client-to-server response, in which each client returns a single scalar score, rather than in a lower end-to-end cost. Under point-to-point query distribution, the total per-query cost is M times that of centralized inference under the stated accounting assumptions, or five times in the evaluated configuration. This overhead can be reduced by broadcast or multicast communication or when the inference sample is already available at the participating clients. Computationally, the local inference cost at client i is $O(|\mathcal{S}_i|d)$ per query and depends mainly on the number of retained support vectors and the feature dimensionality, whereas the server only averages the received scores and applies the calibrated threshold. This allocation of computation suits IIoT deployments in which processing resources are available at or near the data sources, such as edge nodes at wellheads or on production platforms.

Several limitations of the experimental design should be considered when interpreting the comparison between FedRBF-SVM and the centralized RBF-SVM reference. First, the principal comparison was not matched in aggregate training-data availability: the centralized RBF-SVM was trained on k training patterns, whereas FedRBF-SVM trained M local models on a total of $M \times k$ client-level observations, of which fewer may be unique because the independently sampled client subsets could overlap. This difference favors FedRBF-SVM and may contribute to its higher MCC and, at large k, its higher F_1-score relative to the centralized reference.

Second, the principal comparison was not matched in operating point: the centralized RBF-SVM retained its conventional threshold of $\tau = 0$, whereas FedRBF-SVM used validation-calibrated thresholds for target FPRs $\alpha \in \{0.01, 0.03, 0.05\}$. As a result, the centralized reference operated at realized FPRs of approximately 0.30–0.52, compared with at most 0.073 for FedRBF-SVM. This difference favors the centralized reference on recall and FNR but disadvantages it on precision and FPR; therefore, neither the recall gap at small k nor the precision and FPR advantages of FedRBF-SVM reflect a comparison under equivalent operating conditions.

The benchmark analysis in Section 5.4 addresses both limitations: it applied the same validation-based threshold-calibration procedure to all configurations and included a centralized RBF-SVM trained on the full nominal aggregate of $5k$ training patterns. However, it was conducted only at $k = 1000$ and $\alpha = 0.05$ with fixed hyperparameters and should be extended to additional values of k and target FPRs.

Third, the clients represent conceptual well-side processing nodes rather than data partitions corresponding to individual physical wells. Because all client subsets were drawn from the same training subset, the client distributions were effectively independent and identically distributed (IID), and the experiments therefore do not evaluate cross-well statistical heterogeneity or realistic non-IID client distributions. Under these conditions, part of the benefit of decision-score aggregation may reflect the variance reduction typical of ensemble averaging. Future work should accordingly incorporate physical-well-level and explicitly heterogeneous client partitions.

FedRBF-SVM can also be compared with existing distributed and federated SVM-based anomaly-detection methods. Miao et al. [31] proposed a distributed online one-class SVM (OCSVM) for network-based anomaly detection that uses kernel approximation and decentralized stochastic-gradient optimization to enable online learning without requiring participating nodes to exchange raw observations. Their approach was evaluated using synthetic and real datasets and was reported to achieve high detection rates and favorable computational and memory requirements. Anaissi et al. [23] proposed a personalized federated OCSVM (FedP-OCSVM) for structural health monitoring and reported an F_1-score of approximately 0.96 ± 0.02 for damage detection on a cable-stayed bridge dataset. More recently, Frasson and Malchiodi [30] proposed Ensemble SVDD and Support Vector Election, which adapt one-class support-vector methods, including support vector data description (SVDD), to federated settings. The reported results of these studies are not directly comparable with those of the present study because the corresponding approaches use different datasets, anomaly formulations, feature representations, client distributions, and distributed learning procedures. Nevertheless, these studies illustrate the broader applicability of support-vector-based anomaly detection in distributed environments. FedRBF-SVM differs from these approaches by formulating anomaly detection as a supervised binary classification problem using the 3W oil-well dataset, retaining independently trained RBF-SVM models at the participating clients, and performing collaborative classification through the aggregation of decision-function scores rather than exchanging raw training data or local model parameters. In exchange, FedRBF-SVM requires labeled abnormal samples at each

client, whereas one-class methods such as FedP-OCSVM can construct their models from normal-state data alone [23].

7. Conclusions

This study proposed FedRBF-SVM, a federated RBF-SVM approach for collaborative anomaly classification in distributed upstream oil and gas well-monitoring environments. Each client independently trains and retains a local RBF-SVM, and the server averages the local decision-function scores and applies a validation-calibrated threshold, enabling collaborative classification without centralizing local training data or trained models.

FedRBF-SVM was evaluated on the 3W dataset using different numbers of locally stored training patterns (k) per client and validation-calibrated target FPR operating points. Increasing k generally improved the F_1-score, recall, and MCC and reduced the FNR. Precision remained high (0.94–0.99) on the abnormal-skewed test partition, and the realized FPR remained close to each target, with the largest deviation at $\alpha = 0.05$ (at most 0.073). At $k = 2500$ and $\alpha = 0.05$, FedRBF-SVM achieved an F_1-score of approximately 0.79, a precision of approximately 0.97, and a recall of approximately 0.67.

In the benchmark analysis ($k = 1000$, $\alpha = 0.05$), FedRBF-SVM achieved higher mean F_1-score, recall, and MCC than a single local RBF-SVM at a similar mean FPR, whereas the centralized RBF-SVM trained on the nominal aggregate training data outperformed FedRBF-SVM on these metrics. Because the client data in this study were IID, part of the gain over the single model may reflect ensemble averaging. FedRBF-SVM should therefore be regarded as a distributed alternative for collaborative inference in settings where local training data and models cannot be centralized, rather than a replacement for unconstrained centralized training.

The proposed approach preserves training-data and model locality by retaining raw local datasets and trained RBF-SVM models at the participating clients. However, this locality does not constitute a formal privacy guarantee. The collaborative prediction procedure requires the d-dimensional query vector to be made available to the participating clients, which may expose sensitive operational information when wells belong to mutually untrusted operators or administrative domains. Repeated transmission of exact real-valued decision-function scores may also reveal information about the local classifiers and expose them to model-extraction or membership-inference attacks. Consequently, the current formulation provides neither inference-sample confidentiality across mutually untrusted domains nor formal protection against inference-time leakage from the returned scores. Cross-organizational deployment would require an explicit trust model or additional privacy-preserving mechanisms to protect the query vector and local decision scores during collaborative prediction.

Future work will evaluate FedRBF-SVM on physical-well-level partitions that reflect non-IID cross-well heterogeneity, as well as under varying client participation, noisy and missing sensor measurements, and dynamic network conditions. The benchmark comparison will also be extended to additional values of k and target FPRs and to a centralized reference calibrated to the same targets. These evaluations will clarify the relative contributions of aggregate training-data availability and decision-score aggregation to anomaly-classification performance.

Future studies will also investigate alternative aggregation strategies, including weighted averaging, score normalization, averaging of Platt-scaled probability estimates, and majority voting. These strategies will be used to assess whether score calibration and adaptive client weighting improve robustness when clients have heterogeneous local data. Finally, hierarchical and in-network score-aggregation mechanisms will be examined to

evaluate the scalability and communication efficiency of FedRBF-SVM in geographically dispersed well-monitoring deployments.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The 3W dataset (version 2.0.0) analyzed in this study is publicly available at https://github.com/petrobras/3W (accessed on 20 June 2026). The simulation code supporting the findings of this study is available from the corresponding author upon reasonable request.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Adumene, S.; Ikue-John, H. Offshore System Safety and Operational Challenges in Harsh Arctic Operations. *J. Saf. Sci. Resil.* **2022**, *3*, 153–168. [CrossRef]
2. Brkić, D.; Stajić, Z. Offshore Oil and Gas Safety: Protection against Explosions. *J. Mar. Sci. Eng.* **2021**, *9*, 331. [CrossRef]
3. Bello, O.; Holzmann, J.; Yaqoob, T.; Teodoriu, C. Application of Artificial Intelligence Methods in Drilling System Design and Operations: A Review of the State of the Art. *J. Artif. Intell. Soft Comput. Res.* **2015**, *5*, 121–139. [CrossRef]
4. Wanasinghe, T.R.; Wroblewski, L.; Petersen, B.K.; Gosine, R.G.; James, L.A.; De Silva, O.; Mann, G.K.I.; Warrian, P.J. Digital Twin for the Oil and Gas Industry: Overview, Research Trends, Opportunities, and Challenges. *IEEE Access* **2020**, *8*, 104175–104197. [CrossRef]
5. Desai, J.N.; Pandian, S.; Vij, R.K. Big Data Analytics in Upstream Oil and Gas Industries for Sustainable Exploration and Development: A Review. *Environ. Technol. Innov.* **2021**, *21*, 101186. [CrossRef]
6. Fernandes, W., Jr.; Komati, K.S.; Assis de Souza Gazolli, K. Anomaly Detection in Oil-Producing Wells: A Comparative Study of One-Class Classifiers in a Multivariate Time Series Dataset. *J. Pet. Explor. Prod. Technol.* **2024**, *14*, 343–363. [CrossRef]
7. Gil, P.; Martins, H.; Januário, F.; Santos, A.; Palma, L.; Cardoso, A.; Henriques, J. Data Anomaly Detection in Wireless Sensor Networks with Application to an Oil Refinery. In *Proceedings of the 2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO), Ponta Delgada, Portugal, 4–6 June 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 425–429. [CrossRef]
8. He, X.; Robards, E.; Gamble, R.F.; Papa, M. Anomaly Detection Sensors for a Modbus-Based Oil and Gas Well-Monitoring System. In *Proceedings of the 2019 2nd International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 28–30 June 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8. [CrossRef]
9. Benayoune, A. Factors Influencing Industry 4.0 Implementation in Oil and Gas Sector: Empirical Study from a Developing Economy. *Acad. Strateg. Manag. J.* **2021**, *20*, 1–15.
10. Lu, H.; Guo, L.; Azimi, M.; Huang, K. Oil and Gas 4.0 Era: A Systematic Review and Outlook. *Comput. Ind.* **2019**, *111*, 68–90. [CrossRef]
11. Baqer, M. A Machine Learning-Centric Taxonomy and Structured Characterization of Public Datasets for Upstream Oil and Gas. *Big Data Cogn. Comput.* **2026**, *10*, 188. [CrossRef]
12. Jia, Z.; Wang, J.; Deng, C. IIoT-Based Predictive Maintenance for Oil and Gas Industry. In *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering (EITCE 2022), Virtual, China, 21–23 October 2022*; ACM: New York, NY, USA, 2022; pp. 432–436. [CrossRef]
13. Zhang, T.; Gao, L.; He, C.; Zhang, M.; Krishnamachari, B.; Avestimehr, A.S. Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities. *IEEE Internet Things Mag.* **2022**, *5*, 24–29. [CrossRef]
14. Liu, B.; Lv, N.; Guo, Y.; Li, Y. Recent Advances on Federated Learning: A Systematic Survey. *Neurocomputing* **2024**, *597*, 128019. [CrossRef]
15. Baqer, M. Energy-Efficient Federated Learning for Internet of Things: Leveraging In-Network Processing and Hierarchical Clustering. *Future Internet* **2025**, *17*, 4. [CrossRef]
16. Baqer, M. Lightweight Federated Learning Approach for Resource-Constrained Internet of Things. *Sensors* **2025**, *25*, 5633. [CrossRef] [PubMed]
17. Verma, P.K.; Verma, R.; Prakash, A.; Agrawal, A.; Naik, K.; Tripathi, R.; Alsabaan, M.; Khalifa, T.; Abdelkader, T.; Abogharaf, A. Machine-to-Machine (M2M) Communications: A Survey. *J. Netw. Comput. Appl.* **2016**, *66*, 83–105. [CrossRef]
18. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 15. [CrossRef]
19. Pimentel, M.A.F.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A Review of Novelty Detection. *Signal Process.* **2014**, *99*, 215–249. [CrossRef]
20. Hodge, V.J.; Austin, J. A Survey of Outlier Detection Methodologies. *Artif. Intell. Rev.* **2004**, *22*, 85–126. [CrossRef]

21. Martí, L.; Sanchez-Pi, N.; Molina, J.M.; Garcia, A.C.B. Anomaly Detection Based on Sensor Data in Petroleum Industry Applications. *Sensors* **2015**, *15*, 2774–2797. [CrossRef] [PubMed]
22. Vaz Vargas, R.E.; de Melo Junior, A.J.; de Campos Lima, C.B.; de Araújo, J.C.D.; de Araujo Fernandes, M.; Alves Pinto, R.L.; Barbosa, J.R., Jr.; Munaro, C.J.; Ciarelli, P.M.; Varejão, F.M.; et al. 3W Dataset 2.0.0: A Realistic and Public Dataset with Rare Undesirable Real Events in Oil Wells. *Sci. Data* **2026**, *13*, 949. [CrossRef] [PubMed]
23. Anaissi, A.; Suleiman, B.; Alyassine, W. A Personalized Federated Learning Algorithm for One-Class Support Vector Machine: An Application in Anomaly Detection. In *Computational Science—ICCS 2022*; Groen, D., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13353, pp. 373–379. [CrossRef]
24. Wong, P.; Wong, W.K.; Juwono, F.H.; Gopal, L.; Yusoff, M.A. A Minimalist Approach for Detecting Sensor Abnormality in Oil and Gas Platforms. *Petrol. Res.* **2022**, *7*, 177–185. [CrossRef]
25. Vaz Vargas, R.E.; Munaro, C.J.; Ciarelli, P.M.; Medeiros, A.G.; do Amaral, B.G.; Barrionuevo, D.C.; de Araújo, J.C.D.; Ribeiro, J.L.; Magalhães, L.P. A Realistic and Public Dataset with Rare Undesirable Real Events in Oil Wells. *J. Pet. Sci. Eng.* **2019**, *181*, 106223. [CrossRef]
26. Alharbi, B.; Liang, Z.; Aljindan, J.M.; Agnia, A.K.; Zhang, X. Explainable and Interpretable Anomaly Detection Models for Production Data. *SPE J.* **2022**, *27*, 349–363. [CrossRef]
27. Peng, W.; Gao, J.; Chen, Y.; Wang, S. Bridging Data Barriers among Participants: Assessing the Potential of Geoenergy through Federated Learning. *Appl. Energy* **2024**, *367*, 123293. [CrossRef]
28. Chen, B.; Zeng, X.; Zhang, W.; Fan, L.; Cao, S.; Zhou, J. Knowledge Sharing-Based Multi-Block Federated Learning for Few-Shot Oil Layer Identification. *Energy* **2023**, *283*, 128406. [CrossRef]
29. Chen, B.; Zeng, X.; Zhang, W.; Li, Z.; Hou, Z.; Xu, S.; Sun, C.; Han, D. Federated Learning for Cross-Block Oil-Water Layer Identification. *arXiv* **2021**, arXiv:2112.14359. [CrossRef]
30. Frasson, M.; Malchiodi, D. Support Vector Based Anomaly Detection in Federated Learning. In *Engineering Applications of Neural Networks, Proceedings of the 25th International Conference on Engineering Applications of Neural Networks (EANN 2024)*; Springer: Cham, Switzerland, 2024; pp. 274–287. [CrossRef]
31. Miao, X.; Liu, Y.; Zhao, H.; Li, C. Distributed Online One-Class Support Vector Machine for Anomaly Detection over Networks. *IEEE Trans. Cybern.* **2019**, *49*, 1475–1488. [CrossRef] [PubMed]
32. Widodo, A.; Yang, B.-S. Support Vector Machine in Machine Condition Monitoring and Fault Diagnosis. *Mech. Syst. Signal Process.* **2007**, *21*, 2560–2574. [CrossRef]
33. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Agüera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 20–22 April 2017*; Singh, A., Zhu, J., Eds.; PMLR: Cambridge, MA, USA, 2017; Volume 54, pp. 1273–1282. Available online: https://proceedings.mlr.press/v54/mcmahan17a.html (accessed on 27 June 2026).
34. Nicholson, A.; Webber, S.; Dyer, S.; Patel, T.; Janicke, H. SCADA Security in the Light of Cyber-Warfare. *Comput. Secur.* **2012**, *31*, 418–436. [CrossRef]
35. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.-C.; Yang, Q.; Niyato, D.; Miao, C. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
36. He, C.; Annavaram, M.; Avestimehr, S. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Article 1180, 13p.
37. Niknam, S.; Dhillon, H.S.; Reed, J.H. Federated Learning for Wireless Communications: Motivation, Opportunities, and Challenges. *IEEE Commun. Mag.* **2020**, *58*, 46–51. [CrossRef]
38. Tahir, M.; Ali, M.I. On the Performance of Federated Learning Algorithms for IoT. *IoT* **2022**, *3*, 273–284. [CrossRef]
39. Shaheen, M.; Farooq, M.S.; Umer, T.; Kim, B.-S. Applications of Federated Learning: Taxonomy, Challenges, and Research Trends. *Electronics* **2022**, *11*, 670. [CrossRef]
40. Hu, K.; Li, Y.; Xia, M.; Wu, J.; Lu, M.; Zhang, S.; Weng, L. Federated Learning: A Distributed Shared Machine Learning Method. *Complexity* **2021**, *2021*, 8261663. [CrossRef]
41. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
42. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
43. Hsu, C.-W.; Chang, C.-C.; Lin, C.-J. *A Practical Guide to Support Vector Classification*; Technical Report; Department of Computer Science, National Taiwan University: Taipei, Taiwan, 2003.
44. Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M.K.; Ristenpart, T. Stealing Machine Learning Models via Prediction APIs. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016*; USENIX Association: Berkeley, CA, USA, 2016; pp. 601–618.

45. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 3–18. [CrossRef]
46. Claesen, M.; De Smet, F.; Suykens, J.A.K.; De Moor, B. Fast Prediction with SVM Models Containing RBF Kernels. *arXiv* **2014**, arXiv:1403.0736. [CrossRef]
47. Python Software Foundation. Python 3.13.3. Available online: https://www.python.org/downloads/release/python-3133/ (accessed on 27 June 2026).
48. Petrobras. 3W Project: Dataset and Toolkit. Available online: https://github.com/petrobras/3W (accessed on 27 June 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.