Subsea Valve Anomaly Detection: A Comparative Analysis of One-Class Deep Neural Network Architectures

Lesly Shessira Ttito Ugarte[1], **Flavia Bernardini**[1], **Eduardo Andrade**[1], **Felipe Murad**[2], **Esteban Gonzalez Clua**[1]

[1] Institute of Computing – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brazil

[2]Mindsim – Rio de Janeiro – RJ – Brazil
https://www.mindsim.com.br/

leslyugarte@id.uff.br, {fcbernardini,eandrade}@ic.uff.br

fmurad@mindsim.com.br, esteban@ic.uff.br

Abstract. *The subsea valves used in oil and gas extraction processes operate under extreme conditions, and anticipating their anomalies raises important social, technical, and economic implications. This paper proposes a framework based on the Deep Neural Networks for One-Class Classification paradigm on multivariate time series to detect rare anomalies. Based on literature reviews, we designed our framework and observed that our adopted paradigm was not widely adopted, as only one model type is typically used. We evaluate three DNN architectures (CNNs, LSTMs, and Informer) on the public 3W dataset. The empirical results show that the Informer architecture achieves the best results in 4 out of 6 scenarios and the second-best in the other 2.*
Keywords: *Anomaly Detection, Subsea Valve, Deep Neural Network, One Class Classifiers, Time Series*

1. Introduction

While the global community prioritizes green energy, the oil and gas industry remains a primary macroeconomic driver in developing economies, requiring a dual strategy that expands renewables while minimizing extraction risks. In Brazil, the world's 9th-largest oil producer, offshore deepwater and ultra-deepwater presalt fields demand extreme operational efficiency. As part of the Industry 4.0 shift toward smart manufacturing networks, subsea valves serve as a critical frontline defense for surface production chokes and well structural integrity.

Historically, major operators like Petrobras relied on manual visual inspection and isolated acoustic reports for subsea monitoring [Hausler et al., 2015]. However, subsea valves continuously operate under extreme pressure, severe thermal gradients, and harsh chemical environments, leading to accelerated deterioration. Unplanned maintenance or catastrophic failures cause severe economic loss, environmental damage, and prolonged operational downtime [Theyab and Theyab, 2018]. Automating the early detection of subtle operational deviations before they escalate is therefore a critical engineering priority.

Detecting anomalous valve behavior from real-world subsea telemetry presents significant machine learning challenges. Sensor streams consist of noisy, continuous,

multivariate time series with strong temporal dependencies and missing values [Lian et al., 2023]. Furthermore, these systems suffer from extreme class imbalance: normal operations comprise over 99% of data, while structural anomalies remain rare and unmodeled due to conservative maintenance strategies [Vargas et al., 2019]. Because traditional supervised classifiers require well-sampled, balanced representations of failure modes, they fail when applied to real-world operational telemetry.

To address these limitations, this study proposes a temporal anomaly detection framework for subsea valves based on One-Class Classification (OCC) using deep neural network (DNN) architectures. Trained exclusively on normal operational data, OCC models establish baseline behavioral boundaries and flag structural deviations as anomalies. Existing literature rarely evaluates diverse temporal DNN architectures under an OCC setup, particularly in the Oil & Gas domain. To bridge this gap, our framework systematically compares three temporal architectures: a 1D Convolutional Neural Network (CNN) for spatial-temporal feature extraction [LeCun et al., 1989], a Long Short-Term Memory (LSTM) network for sequential dependencies [Hochreiter and Schmidhuber, 1997], and the Informer architecture for long-term multi-scale tracking [Zhou et al., 2021]. This framework was developed and validated across four main phases:

1. Literature Synthesis & Gap Identification: A systematic literature review mapped machine learning applications in industrial valves, while a narrative review focused on time-series OCC via DNNs to establish theoretical novelty.
2. Data Acquisition: To resolve the total absence of public subsea valve datasets, we utilized the open-source Petrobras 3W Dataset [Vargas et al., 2019], which provides authentic, simulated, and hand-drawn well anomaly instances.
3. Framework Design: A reproducible data engineering pipeline applied specialized preprocessing and sliding-window transformations for deep temporal learning.
4. Empirical Validation: Hyperparameter grid searches and performance evaluations using Precision, Recall, and F1-score to identify operational bounds across architectures. The reported metrics are from the best model obtained from diverse training runs.

The remainder of this paper is structured as follows: Section 2 covers theoretical background on multivariate time series, OCC, and DNN architectures. Section 3 details the literature review and research gaps. Section 4 presents the proposed methodological framework, while Section 5 describes the Petrobras 3W dataset. Section 6 details the experimental setup and hyperparameter tuning. Finally, Sections 7 and 8 present the results, discussion, and conclusions with future work.

2. Theoretical Background

Multivariate Time Series T is defined as a sequence of n real-valued observations ordered across uniform chronological intervals. Mathematically, it is represented by $Y_t \in \mathbb{R}^g; \quad t = 1, 2, \ldots, n$, where g is the spatial dimensionality of the observations, corresponding to the total number of telemetry sensors monitoring the asset concurrently. In our subsea valve scenario, each observation vector captures physical parameters such as temperature, pressure, and volumetric flow metrics.

One-Class Classification (OCC) models address contexts where the target anomalous class is completely absent, poorly sampled, or structurally unbounded during the model

optimization phase [Khan and Madden, 2014]. Learning an efficient classifier is accomplished by defining a boundary strictly encapsulating the known positive (valves' normal behavior) class space. Given a training distribution $X_{\text{tr}} = \{Y_t\}_{t=1}^{m}$ representing exclusively normal operation, the model learns a mapping function $f(Y) \rightarrow \{0, 1\}$ defined by Eq. 1, which allows the system to remain highly robust against unmodeled and entirely novel failure modes since any significant deviation from the expected normal signature triggers an anomaly alarm.

$$f(Y) = \begin{cases} 0 & \text{if } Y \in \text{Normal Boundary} \\ 1 & \text{if } Y \notin \text{Normal Boundary (Anomaly)} \end{cases} \tag{1}$$

DNN Architectures: We adopted the following three models in our framework:

- *CNN:* 1D CNNs apply discrete spatial-temporal convolutions along the temporal axis of the sliding window matrix [Griffiths, 2021]. The network uses localized filter kernels to automatically extract shift-invariant spatial-temporal feature maps, avoiding manual engineering of statistical features.
- *LSTM:* A specialized Recurrent Neural Network (RNN) architecture engineered to track long-range sequential dependencies across time-series streams while mitigating the vanishing gradient problem. Instead of static representations, an LSTM block utilizes an internal cell state regulated by three distinct gating mechanisms: input, forget, and output gates. These continuous mechanisms allow the network to dynamically retain historical telemetry context, discard irrelevant noise, and selectively propagate critical sequential patterns across long temporal horizons.
- *Informer:* A transformer-based model designed for long-sequence multivariate time-series processing [Zhou et al., 2021]. It introduces the *ProbSparse* Self-Attention mechanism, which reduces the standard quadratic computational complexity $\mathcal{O}(L^2)$ of self-attention down to an efficient $\mathcal{O}(L \ln L)$ by selecting dominant queries based on a Kullback-Leibler divergence measurement strategy. This structural property allows it to extract global dependencies across large temporal horizons without suffering from standard memory or gradient constraints.

Evaluation Metrics: To calculate the evaluation metrics, one must consider a Confusion Matrix of a binary classification problem, generated on a test set of N instances, composed by the values TP (True-Positives), being the anomalies classified as anomalous operation; TN (True-Negatives), being the normal instances classified as normal operation; FP (False-Positives), being the normal instances classified as anomalous operation; FN (False-Negatives), being the anomalies instances classified as normal operation. To evaluate performance beyond accuracy ($Acc = (TP + FN) \div (N)$, we employ other metrics derived from the standard Confusion Matrix: **Precision** ($Prec = TP \div (TP + FP)$), **Recall (Sensitivity)** ($Rec = TP \div (TP + FN)$), and **F1-Score** ($F1 = 2 \times (Prec \times Rec) \div (Prec + Rec)$), which is the harmonic mean balancing precision and recall performance.

3. Literature Review

To establish our experimental design, we conducted a Systematic Literature Review (SLR) following the PRISMA methodology across the ACM Digital Library, IEEE Xplore,

and Scopus to identify machine learning algorithms historically used for industrial valve anomaly detection. The full protocol and foundational analysis were published in <<**Omitted for Double Blind Review**>>.

The SLR revealed a critical reproducibility deficit: only 5 of 23 scrutinized global studies shared underlying telemetry or open benchmarks, none of which targeted subsea production valves. Mechanistically, the literature falls into three paradigms: non-deep statistical learners, supervised shallow/deep neural networks, and alternative machine learning frameworks. Deep supervised architectures achieved the highest predictive accuracies due to their capacity to model temporal correlations. Notably, hybrid 1D CNN-LSTMs have been used to isolate mechanical misalignments in cooling systems [Zhao et al., 2021] and PCA-Informer hybrids have successfully captured long-range acoustic and vibration signals in nuclear valve systems [An et al., 2022].

These findings justified discarding Multi-Layer Perceptrons (MLPs) – which treat data points independently – in favor of deep temporal models. Deep learning automatically extracts hierarchical features directly from raw, noisy, multivariate telemetry without manual feature engineering. Specifically, 1D CNNs isolate localized spatial and cross-metric patterns; LSTMs govern sequential dependencies via regulatory gating; and attention-based Informers scale efficiently to capture long-range global relationships.

However, translating these models to subsea operations revealed a major structural obstacle: in the single publicly available dataset we found, there is significant data scarcity and class imbalance across the 21 wells of the open-source Petrobras 3W Dataset [Vargas et al., 2019]. Because many wells contain few or no actual anomaly instances, supervised training is unfeasible. This motivated us to conduct a narrative review of One-Class Classification (OCC) for oil and gas maintenance. While OCC frameworks exist in this area, the current time-series literature typically evaluates them on a single DNN type. To our knowledge, no prior work has compared CNN, LSTM, and Informer architectures within a unified OCC framework for subsea valve monitoring through multivariate time series. Our framework diverges from existing upstream oil and gas telemetry literature in paradigm, scalability, and architectural efficiency in the following aspects:

Paradigm Shift: Unlike supervised or semi-supervised deep architectures that require predefined fault labels or known structural degradation signatures[Aranha et al., 2024], our framework trains exclusively on normal baseline process parameters, allowing it to flag unseen, rare anomalies autonomously.

Dimensionality and Scalability: alternative OCC time-series approaches rely on traditional methods like K-means clustering with Dynamic Time Warping (DTW) Barycenter Averaging and the Apriori algorithm [Machado et al., 2025], although DTW introduces severe computational complexities that fail to scale to high-frequency subsea streams, and its reliance on aggressive dimensionality reduction (1–3 variables) strips away vital cross-sensor correlations.

In contrast, our framework preserves the full multivariate space, utilizing a sliding-window technique tailored to continuous, multi-sensor fluid dynamics. By benchmarking 1D CNNs, LSTMs, and the Informer architecture, this work analyzes performance boundaries for capturing complex temporal dependencies across parallel sensor streams, achieving far more efficient capture than standard sequential or DTW alternatives.

4. Our Framework

Our framework follows 5 main steps to construct and evaluate DNNs as OCCs on multivariate time series. Although we aim to process multiple time-series data from subsea valves, our framework is designed to be applicable across diverse industrial IoT and multi-sensor telemetry frameworks.

Step 1 – Standardized Data Preprocessing Protocol: The raw tabular multivariate telemetry undergoes a strict sequence to filter noise and standardize inputs:

1. *Cleaning:* Duplicate timestamps, absolute nulls, unrecorded intervals, structural errors, and unphysical sensor values (e.g., negative pressures) are removed.
2. *Feature Selection:* Uninformative features are identified by analyzing their completeness and variability across the dataset. Features that are fully null, contain invalid values, or remain constant throughout an instance are discarded, as they do not provide information for anomaly identification. The remaining features are retained, including correlated sensor measurements, to preserve the multivariate representation of the well's operational behavior.
3. *Normalization:* Remaining physical features are mapped to a $[0, 1]$ range using Min-Max scaling based on normal operational ranges.

Step 2 – Sliding Window Temporal Transformation is used to encode sequential dependencies and structural temporal correlations into the DNNs. First, construct a window-matrix dataset WM with the continuous telemetry streams of total row count N using a fixed sliding window horizon W with a step size of 1 – $WM = DS[i : i + W]$. Then, structure the extracted sub-matrix blocks into a continuous 3D feature tensor shape: $[\text{Samples}, WM, \text{Features}]$.

Step 3 – Final Datasets Construction:

1. *Sequential Partitioning:* Construct $DS_0, DS_1, ..., DS_L$ datasets, being DS_0 only with normal data and $1, ...L$ the abnormal events. So, $DS_1, ..., DS_L$ contains data from the L different types of anomalies or abnormal events found in the analyzed telemetry. As our main target is telemetry from subsea wells, the telemetry will be from a single well. Then, construct $DS^+_{E_1}, ..., DS^+_L$ from merging DS_0 with each $DS_1, ..., DS_L$, that we call processed datasets, having approximately similar sizes of instances, and also chronologically constructing, considering a 70/30 partition for training and test sets of each processed dataset. The training dataset will contain only 100% of normal operational regimes for all processed datasets, and the test set will contain a mixture of normal and abnormal instances. For this, the 3D tensor is split chronologically into a 70/30 partition.
2. *Labeling:* Because there are too few abnormal events (the dataset we explored has 7) and due to imbalance, our approach constructs one OCC for normal behavior, treating all abnormal events as anomalies. So, the target variable is mapped as Normal = 0 and Anomaly = 1.

Step 4 – DNNs Construction for OCC: Three distinct DNN temporal architectures are implemented to capture feature relationships across the five retained industrial sensor streams:

- *1D CNN:* Built with two convolutional layers, two MaxPooling layers, a Flatten layer, and two Dense layers (128 neurons) with a 0.2 dropout rate.

- *LSTM:* Configured for sequential dependency tracking via gating mechanisms.
- *Informer:* A specialized Transformer variant optimized for long-term time-series forecasting and multi-scale feature tracking.
- *Outputs:* All architectures terminate with a Sigmoid activation layer to output a reconstruction error threshold mapped to binary operational states.
- *Optimization:* The models must be trained using hyperparameter tuning.

Step 5 – Empirical Framework Evaluation: The final step establishes operational performance boundaries on the dataset under scrutiny:

- *Inference* is executed over the remaining 30% sequential test blocks containing real, simulated, and hand-drawn anomalies.
- *Classification performance indices* are computed – Acc, $Prec$, Rec, and $F1$.
- *Computational environments* leverage Python 3.10, TensorFlow 2.14, PyTorch 2.1, and Scikit-learn 1.1 running on dedicated GPU infrastructure.

5. The 3W Dataset

The (benchmark) 3W dataset contains multivariate time-series telemetry designed to capture diverse operational anomalies in naturally flowing offshore oil wells. The complete dataset comprises 1,984 CSV files from 21 distinct wells ($W_1, ..., W_{21}$), originally organized into folders for normal and various abnormal events. The events are: E_0 Normal; E_1 Abrupt Increase of BSW; E_2 Spurious Closure of DHSV; E_3 Severe Slugging; E_4 Flow Instability; E_5 Rapid Productivity Loss; E_6 Quick Restriction in PCK; E_7 Scaling in PCK; E_8 Hydrate in Production Line. Table 1 shows the number of CSV files per type of event ($E_0, ..., E_8$) and by type of data (real data, simulated data, and hand-generated data). In this work, we only used the real data. We can observe that all abnormal events (or anomalies) have scarce CSV files, except for E_4.

Table 1. Number of CSV files by type of events and type of data, considering the entire dataset (all the wells). Ab stands for abnormal events ($E_1, ..., E_8$)

Type of Event	Real	Sim.	Hand	Total
E_0 **(Normal)**	**597**	-	-	**597**
E_1	5	114	10	129
E_2	22	16	-	38
E_3	32	74	-	106
E_4	344	-	-	344
E_5	12	439	-	451
E_6	6	215	-	221
E_7	4	-	10	14
E_8	3	81	-	84
Subotal (**Ab**)	428	939	20	1397
Total	**1025**	**939**	**20**	**1984**

Following the physical path of the fluid dynamics from the reservoir to the surface, the eight instrumentation sensors are arranged in the following sequence: fluid pressure is first captured deep within the wellbore by the **P-P** (Pressure at the Permanent Downhole Gauge), before traveling up to the seabed where it passes the subsea Christmas tree monitored by the co-located **P-T** (Pressure at the TPT) and **T-T** (Temperature at the TPT). The

flow then progresses into the production manifold, where its state is measured immediately before the choke valve by **P-M** (Pressure upstream of the PCK)[1] and immediately after by **T-J-C** (Temperature downstream of the PCK). Finally, the artificial gas lift injection line—which assists the primary fluid flow—is monitored by **QGL** (Gas lift flow rate) before the gas passes through its respective choke valve, where its final parameters are captured by **P-J** (Pressure downstream of the GLCK) and **T-J-L** (Temperature downstream of the GLCK). Table 2 shows an example of a CSV file, where we can observe that each file has millions of entries.

Table 2. Example of a CSV file

	timestamp	P-P	P-T	T-T	P-M	T-J-C	P-J	T-J-L	OGL	class
0	2012/04/10 19:23	2.157	684335400.0	59.245	1625144.0	NaN	1.552	NaN	0.06	0.0
1	2012/04/10 19:23	2.157	684189700.0	59.3	1623155.0	NaN	1.552	NaN	0.06	0.0
2	2012/04/10 19:23	2.157	684043900.0	59.3	1621166.0	NaN	1.552	NaN	0.06	0.0
...										
69123	2018/06/21 13:25	-1.180	21228450.0	117.3	12311950.0	64.73	2.843	NaN	0.00	1.0
69124	2018/06/21 13:25	-1.180	21228560.0	117.3	12312070.0	64.73	2.843	NaN	0.00	1.0
69125	2018/06/21 13:25	-1.180	21228670.0	117.3	12312190.0	64.73	2.843	NaN	0.00	1.0
...										

(14044002 rows × 10 columns)

Each instrumentation sensor generates a time series, which could also be called a parameter or variable. It's worth noticing that the raw data exhibits real-world constraints: 31.17% of the variables are entirely missing due to network dropouts, 9.67% are frozen due to system configuration failures, and a small fraction of observations remain unlabeled. To provide spatial context, Figure 1 illustrates the physical placement of these instrumentation sensors across the different stages of the subsea oil extraction process.

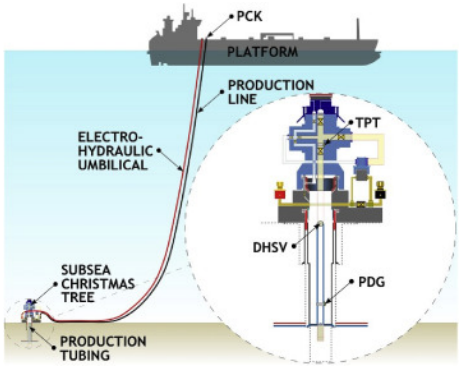


Figure 1. Structural schematic of a subsea production well and its sensor telemetry infrastructure [Harrouz et al., 2022].

6. Instantiating our framework on 3W dataset

Our framework's execution begins with the data preprocessing protocol (Step 1) that addresses the inherent structural imperfections of the 3W dataset. To establish a baseline for OCC, our experimental scope was limited to W_2, which contains the largest volume of

[1]It is worth noting that this specific architectural layout may vary across different offshore production platforms. In certain configurations, a subsea manifold is completely absent, routing the fluid stream directly from the well to the Floating Production Storage and Offloading (FPSO) unit; conversely, other production fields may incorporate a cluster of two or more interconnected subsea manifolds.

normal baseline instances (210 CSV files), along with representative anomalies or events (1 CSV file each for E_1 and E_2, and 113 CSV files for E_6). Preprocessing activities include converting transient anomaly-start states (values > 100) into a definitive binary anomaly class (1), dropping invalid negative sensor values, and standardizing the telemetry via Min-Max normalization. Furthermore, high-null (a high proportion of missing values) and constant time series (P-PDG, T-JUS-CKGL, and QGL) are removed, leaving five cross-sensor dimensions: P-TPT, T-TPT, P-MON-CKP, T-JUS-CKP, and P-JUS-CKGL. Linear interpolation is then applied to reconstruct a continuous, chronological 1-second sequence.

In Step 2, we used $W = 400$. In Step 3, the training partition contains only normal, real baseline data from W_2, composing DS_{E_0}. In contrast, the remaining chronological sequences contain both normal and anomalous events E_1, E_2, E_6, being attributed to datasets $DS_{E_1}, DS_{E_2}, DS_{E_6}$, respectively. Then, the datasets $DS^+_{E_1}, DS^+_{E_2}, DS^+_{E_6}$ are constructed with approximately 70,000 data points each. For this, we concatenated data from DS_0 with DS_1, DS_2, DS_6, preserving the timestamps of instances and interleaving normal and abnormal instances. Table 3 shows the instance data quantities of normal (# Normal) and abnormal operation (# Anomaly) distributions. We train and test each DNN (Steps 4 and 5) with each $DS \in \{DS^+_{E_1}, DS^+_{E_2}, DS^+_{E_6}\}$ is used to.

Table 3. Quantity of instances for training Tr and testing Te of each processed dataset

Case	Dataset	# Normal	# Anomaly
$DS^+_{E_1}$	Tr	49,000	0
	Te	21,000	9,613
$DS^+_{E_2}$	Tr	49,000	0
	Te	21,000	5,365
$DS^+_{E_6}$	Tr	46,768	0
	Te	24,370	9,113

In Step 4, the CNN model was built with TensorFlow/Keras, featuring alternating convolutional and MaxPooling layers, coupled with a Sigmoid activation function to map outputs to 0-1. The LSTM network, also built with Keras, uses four stacked recurrent layers with 0.2 dropout to mitigate gradient vanishing. In parallel, to circumvent the classic vanishing gradient and efficiency constraints of traditional sequential networks over long ranges, the Informer model was implemented in PyTorch, requiring the conversion of input variables into dedicated multidimensional tensor signatures. Networks were compiled using a Binary Cross-Entropy loss function and the Adam optimizer (1×10^{-4} learning rate), updating weights via backpropagation over mini-batches (32, 64, or 128) across 5 to 60 epochs (Tab. 4). Systematic hyperparameter tuning was performed across varying configurations, defining our experimental scenarios (Tab. 5): batch sizes from 32 to 128 and epochs from 5 to 60, to ensure optimal convergence. The hardware infrastructure included an AMD Ryzen 5000 processor, 16GB of RAM, and an NVIDIA RTX 6GB GPU. Post-tuning, optimal weights and features were saved as serialized files.

Step 6 involved calculating the set of metrics $\{Acc, Prec, Rec, F1\}$ for each pair of DNN and test set of each processed dataset of the set $\{DS^+_{E_1}, DS^+_{E_2}, DS^+_{E_6}\}$, using Scikit-Learn. Each experimental scenario was executed over several runs, being presented

Table 4. Hyperparameters used for each DNN architecture

Hyperparameters	**CNN**	**LSTM**	**Informer**
Size	Neurons: 128	Layers: 4	Dimensions: 5
Loss Function	Binary Cross Entropy	Binary Cross Entropy	Binary Cross Entropy
Dropout	0.2	0.2	0.1
Optimizer	Adam	Adam	Adam
Learning rate	$1 * 10^{-4}$	$1 * 10^{-4}$	$1 * 10^{-4}$
Batch sizes	$\{64, 128\}$	$\{32, 64\}$	$\{32, 64\}$
Epochs	$\{20, 40, 60\}$	$\{5, 10, 20\}$	$\{5, 10\}$

Table 5. Experimental scenarios (Exp. Scn.) used for each DNN architecture

	CNN		LSTM		Informer	
Exp. Scn.	Batch Sizes	#Epochs	Batch Sizes	#Epochs	Batch Sizes	#Epochs
Exp_1	64	20	32	5	32	5
Exp_2	64	40	32	10	32	10
Exp_3	64	60	32	20	64	5
Exp_4	128	20	64	5	64	10
Exp_5	128	40	64	10	–	–
Exp_6	128	60	64	20	–	–

the result of the best model. All the codes we developed to execute our framework are available at `https://github.com/slarkdev/AnomaliesDetection`.

7. Results and Analysis

Table 6 shows Acc and $F1$ metrics for each experimental configuration of each DNN architecture (CNN, LSTM, Informer) and each dataset ($DS^+_{E_1}, DS^+_{E_2}, DS^+_{E_6}$). Results in bold and marked with * are the best ones for the pair (DNN,DS), and in bold, blue, and marked with ★ are the best ones for the dataset DS. Relying exclusively on the standard Accuracy metric to diagnose multi-sensor anomalies can lead to misleading conclusions due to unbalanced classes. We also generated data visualizations based solely on the $F1$ measure, shown in Fig. 2. We can observe that the Informer architecture outperformed the other models for $DS^+_{E_1}$ (in Exp_1), achieving Acc of 0.997% and $F1$ of 0.996%, and for $DS^+_{E_6}$ (in Exp_4), achieving Acc of 0.929% and $F1$ of 0.909%. Although the CNN architecture achieved the best result for $DS^+_{E_2}$ (Exp_1), with Acc of 0.933% and $F1$ of 0.860%, we observe that the Informer architecture also achieved a good result, with Acc of 0.909% and $F1$ of 0.803%. From our understanding, the Informer's good performance stems from the *ProbSparse* attention mechanism, which enables the network to model global temporal dependencies across long multi-sensor sequences without sacrificing contextual resolution.

It's also worth noting in both the table and the images that all the architectures struggled to achieve good performance on the $DS^+_{E_2}$ and $DS^+_{E_6}$ datasets. Particularly, the evaluation on the $DS^+_{E_6}$ benchmark revealed performance drops across all three neural architectures. Observing Table 3, we can observe that $DS^+_{E_2}$ has less anomaly data than the others, which could explain these results. However, $DS^+_{E_6}$ has a data distribution similar to $DS^+_{E_1}$, so data imbalance does not explain the DNNs' performance. On the other hand, this drop occurred because the underlying instances in $DS^+_{E_6}$ lacked the continuous sequential structure present in $DS^+_{E_1}$, confirming that data continuity is essential for op-

Table 6. Results obtained on test sets of $DS^{+}_{E_1}, DS^{+}_{E_2}, DS^{+}_{E_6}$ for each DNN architecture – CNN, LSTM, Informer.

DNN Arch.	Exp. Scn.	$DS^{+}_{E_1}$		$DS^{+}_{E_2}$		$DS^{+}_{E_6}$	
		Acc	$F1$	Acc	$F1$	Acc	$F1$
CNN	Exp_1	0.986	0.985	**0.933★**	**0.860★**	0.212	0.260
	Exp_2	0.973	0.970	0.924	0.842	0.669	0.564
	Exp_3	0.884	0.858	0.924	0.842	0.587	0.612
	Exp_4	0.986	0.985	0.806	0.694	0.491	0.422
	Exp_5	**0.987***	**0.985***	0.826	0.741	0.698	0.613
	Exp_6	0.984	0.982	0.825	0.741	**0.717***	**0.659** *
LSTM	Exp_1	0.749	0.632	0.830	0.754	0.602	0.508
	Exp_2	0.807	0.739	0.811	0.727	0.540	0.477
	Exp_3	0.808	0.740	0.831	0.755	0.405	0.265
	Exp_4	**0.981***	**0.979***	**0.832***	**0.756***	0.451	0.323
	Exp_5	0.580	0.180	0.828	0.750	0.482	0.361
	Exp_6	0.822	0.189	0.828	0.751	**0.835**	**0.797**
Informer	Exp_1	**0.997★**	**0.996★**	0.906	0.796	0.470	0.398
	Exp_2	0.997	0.996	0.874	0.727	0.449	0.438
	Exp_3	0.997	0.996	0.874	0.727	0.470	0.397
	Exp_4	0.997	0.996	**0.909***	**0.803***	**0.929★**	**0.909★**

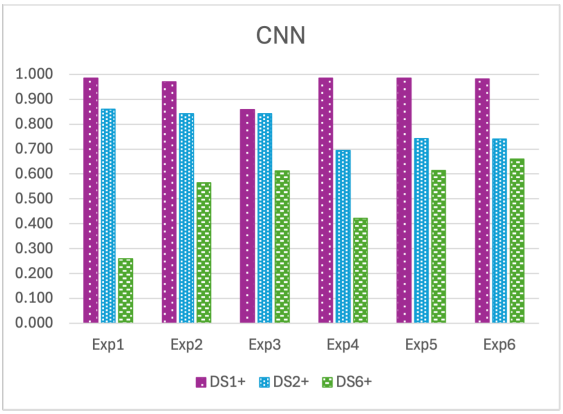
timizing temporal learning models. Another reason is how the data is presented for each type of anomaly. So, a better explanation for these observations is that the concepts of normal behavior and anomaly are more difficult to separate. Still, we consider that we obtained very good results for all the scenarios we explored.

Regarding structural behavior, the CNN model proved to be the most computationally efficient architecture, achieving the fastest training and inference speeds. However, it required extensive threshold adjustments based on Euclidean distance measurements to deliver competitive classification performance. The LSTM network demonstrated stable performance across configurations without extensive hyperparameter tuning, but it required the longest training times due to its sequential recurrence constraints. The Informer model balanced these trade-offs, offering moderate execution speeds while delivering the highest overall detection rates for complex multivariate sequences.

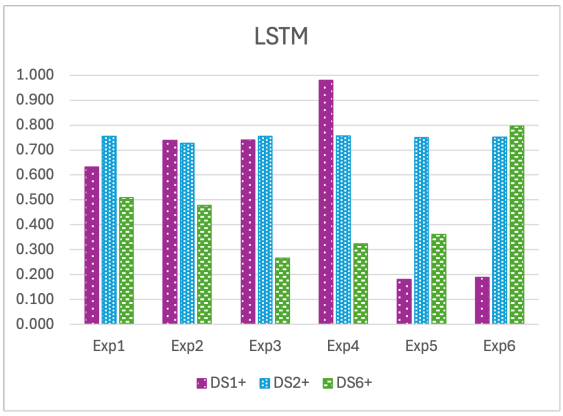
8. Conclusions

This study evaluated a One-Class Classification framework for subsea valve anomaly detection using deep neural networks trained exclusively on normal operational telemetry, effectively overcoming severe industrial class imbalances. Our empirical evaluation on the 3W dataset demonstrated that while CNNs offer computational speed, the Informer architecture achieves a 99.6% F1-score on 1 of 3 anomaly benchmarks. While CNNs offer the fastest execution speeds, they require significant threshold tuning to achieve comparable accuracy. LSTMs deliver reliable performance but are constrained by higher computational latency. We could observe that our framework provides a viable data-driven methodology to mitigate catastrophic failures, minimize environmental contamination risks, and reduce costly stagnation downtime for offshore operations.

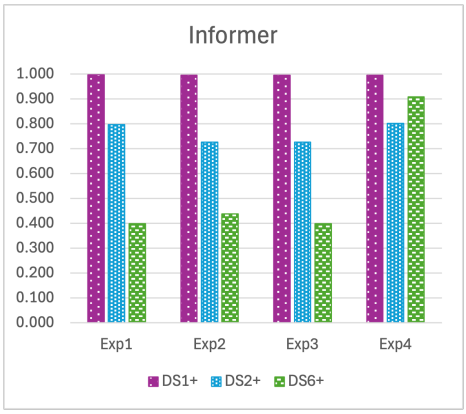
Although we executed diverse runs for each experimental scenario $Exp_1, ..., Exp_6$, we only reported the results for the best model. As CNN, LSTM, and Informer are all stochastic deep learning models, a future work should consider developing a systematic



(a) $F1$ **metrics for DNN CNN.**



(b) $F1$ **metrics for DNN LSTM.**



(c) $F1$ **metrics for DNN Informer.**

Figure 2. $F1$ **metrics**

model-selection and statistical-comparison protocol. Also, future work should also investigate Transfer Learning protocols to adapt these optimized models across new wells featuring distinct physical geometries and sensor configurations.

Statement of AI Use: During the preparation of this work, the author used Gemini and Grammarly to improve the readability, language, and structural flow of the manuscript and to refine the prose. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the final published article. The authors affirm that no generative AI tools were used to create, alter, or manipulate core scientific data, original figures, or primary research findings.

References

An, Z., Cheng, L., Guo, Y., Ren, M., Feng, W., Sun, B., Ling, J., Chen, H., Chen, W., Luo, Y., and Yang, Z. (2022). A novel principal component analysis-informer model for fault prediction of nuclear valves. *Machines*, 10(4).

Aranha, P. E., Lopes, L. G. O., Paranhos Sobrinho, E. S., Oliveira, I. M. N., de Araújo, J. P. N., Santos, B. B., Lima Junior, E. T., da Silva, T. B., Vieira, T. M. A., Lira, W. W. M., Policarpo, N. A., and Sampaio, M. A. (2024). A system to detect oilwell anomalies using deep learning and decision diagram dual approach. *SPE Journal*, 29(03).

Griffiths, I. (2021). Automated condition monitoring using artificial intelligence. In *Proc. 2020 4th Int. Conf. Vision, Image and Signal Processing*, ICVISP 2020. ACM.

Harrouz, A., Toubakh, H., Mohamed Redouane, K., Sayed Mouchaweh, M., and Salem, H. (2022). Self adaptive learning scheme for fault prognosis in oil wells and production and service lines. *Annual Conference of the PHM Society*, 14.

Hausler, R., Krishnamurthy, R., and Sherar, B. W. (2015). Observation of productivity loss in large oil wells due to scale formation without apparent production of formation brine. In *NACE CORROSION*, pages NACE–2015. NACE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Khan, S. and Madden, M. (2014). One-class classification: Taxonomy of study and review of techniques. *The Knowledge Engineering Review*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Lian, Y., Geng, Y., and Tian, T. (2023). Anomaly detection method for multivariate time series data of oil and gas stations based on digital twin and MTAD-GAN. *Applied Sciences*, 13(3).

Machado, A. P. F., Munaro, C. J., and Ciarelli, P. M. (2025). Enhancing one-class classifiers performance in multivariate time series through dynamic clustering: A case study on hydraulic system fault detection. *Expert Systems with Applications*, 286.

Theyab, M. and Theyab, M. A. (2018). Severe slugging control: Simulation of real case study. *J Environ Res*, 2(1):5.

Vargas, R. E. V., Munaro, C. J., Ciarelli, P. M., Medeiros, A. G., do Amaral, B. G., Barrionuevo, D. C., de Araújo, J. C. D., Ribeiro, J. L., and Magalhães, L. P. (2019). A realistic and public dataset with rare undesirable real events in oil wells. *Journal of Petroleum Science and Engineering*, 181:106223.

Zhao, Q., Cheng, G., Han, X., Liang, D., and Wang, X. (2021). Fault diagnosis of main pump in converter station based on deep neural network. *Symmetry*, 13(7).

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35.