UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

APPLICATION OF SUPERVISED CLASSIFICATION AND TIME-SERIES MODELS ON PREDICTION OF UNDESIRABLE EVENTS IN OFFSHORE OIL PRODUCTION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

In partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

Gunash Khankishiyeva

Norman, Oklahoma

2025

APPLICATION OF SUPERVISED CLASSIFICATION AND TIME-SERIES MODELS ON PREDICTION OF UNDESIRABLE EVENTS IN OFFSHORE OIL PRODUCTION

A THESIS APPROVED FOR THE

GALLOGLY COLLEGE OF ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Charles D. Nicholson, Chair

Dr. Hamidreza Karami Mirazizi

Dr. Theodore B. Trafalis

This thesis is dedicated to my parents for their love, support, and encouragement.

Acknowledgment

I want to thank Dr. Charles Nicholson for always being there to guide and support me. His patience, kindness, and dedication to his students have encouraged me to do more than I thought I could. I am fortunate to have had him as an academic supervisor and to have learned so much from him. Dr. Nicholson's guidance and belief in my capabilities have empowered me to surpass my perceived limits and achieve significant milestones during my master's degree.

I am also very thankful to Dr. Hamidreza Karami for his patience, support, and willingness to help with my questions about oil and gas production. His guidance has been invaluable, and he has played a significant role in my research.

Special thanks to Dr. Matthew Beattie for his invaluable contributions, guidance, and provision of crucial data throughout the research process. Dr. Matthew Beattie's expertise and insightful feedback have significantly enriched the quality of this study.

I express my sincere appreciation to Dr. Theodore B. Trafalis, a dedicated committee member, for his timely guidance, feedback, and support. His constructive feedback and corrections have been invaluable in refining the content and structure of this thesis.

Great thanks to the University of Oklahoma and the Gallogly College of Engineering Data Science and Analytics department for providing me the opportunity to learn and grow as a professional data scientist.

To my husband, Orkhan Khankishiyev—thank you for always standing by my side, no matter what. Thank you for being my best friend, partner, and beloved husband for your unwavering support, encouragement, and belief in me throughout this journey

Lastly, I am truly grateful to my parents for their constant love, support, and encouragement throughout my life and education. Your belief in me, even when I doubt myself, has been a steady

source of strength. You've always been there for me every step of the way, offering comfort when things were tough and celebrating with me during the good times. I couldn't have reached this point without your sacrifices, patience, and unwavering faith in me. Thank you for always standing by my side; this achievement is as much yours as it is mine.

Table of Contents

List of Figures

List of Tables

Abstract

Offshore oil production is faced with critical challenges due to rare but high-impact undesirable events that disrupt well operations. The early prediction of events such as sudden water breakthroughs, valve failures, flow instabilities, and blockages is considered essential for preventing production losses, environmental incidents, and safety hazards. Traditional physics-based models are often limited in handling the complex, multivariate nature of these problems, whereas supervised machine learning and time-series modeling are increasingly applied to provide data-driven solutions for detecting subtle patterns that occur before failure.

In this thesis, the application of supervised classification algorithms and temporal models has been investigated for the purpose of predicting undesirable events in offshore oil wells. A publicly available dataset (3W), which contains over 50 million sensor readings from an offshore field with eight documented types of production anomalies, has been used as a benchmark. A detailed study of the production challenges represented in the data has been carried out, including anomalies such as Abrupt Increase of BSW (Basic Sediment and Water), Spurious DHSV closures, Severe Slugging, Flow Instability, Rapid Productivity Loss, Quick Choke Restrictions, Scaling, and Hydrate formation, along with their operational relevance. The structure of the dataset, the origins of the sensor measurements, and the labeling of instances have been reviewed in order to inform feature engineering and model development strategies.

Several supervised learning methods – including k-Nearest Neighbors (KNN), Decision Trees (DT), Random Forests (RF), and Artificial Neural Networks (ANN) – have been implemented and compared. Additional classifiers such as Support Vector Machines (SVM) and ensemble boosting methods have also been used to evaluate classification accuracy and robustness. In parallel, time-series modeling techniques have been applied to capture the temporal

dependencies present in sensor data. Approaches such as recurrent neural networks (e.g. LSTM autoencoders) have been examined for their ability to predict the early stages of events based on time-dependent sensor information.

To support the development of realistic time-series models, simulated datasets have been transformed using autoencoder-based distortion techniques. These distortions have been learned from real well signals and applied to simulated data in order to preserve the dynamic characteristics of operational behavior while introducing realistic anomalies. Recurrent neural network models, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures, have been applied under both multiclass classification and regression-based frameworks. In particular, a regression-based approach has been used to forecast class transitions by treating normal (class 0), transient (class 101), and anomaly (class 1) states as sequential time steps—representing past, current, and future behavior. This structure has been designed to simulate the natural progression of events in the wells. Instead of directly classifying the well state, the model has been trained to predict a numerical value representing the next probable class, which has then been post-processed and mapped back to discrete class labels. By adopting this method, a forward-looking prediction of system behavior has been made possible, offering the potential for earlier anomaly detection.

The findings highlight the advantages of integrating supervised classification with time-series modeling and realistic data transformation. The combination of these methods has contributed to a more complete understanding of offshore well conditions and provided a practical foundation for building intelligent systems capable of monitoring operations and supporting decision-making in oil production environments.

Chapter 1 : INTRODUCTION

1.1 Problem Background

Offshore oil production has been recognized as a technologically intensive activity that is carried out in remote and harsh environments where unexpected incidents may result in severe consequences. The oil and gas industry is recognized as one of the key energy sources in the world, but significant challenges in instrumentation and automation are presented by the sector. Efficiency is considered paramount, but mechanical problems often result in long troubleshooting and repair times (Table 1-1).

Table 1-1. Estimated time windows to confirm undesirable events. (Vargas et al., 2019)

Type of Undesirable Event	Time Window to Confirm the Event
Label 1 – Abrupt increase of Basic Sediment and Water (BSW)	12 h
Label 2 – Spurious closure of the Downhole Safety Valve (DHSV)	5 min -20 min
Label 3 – Severe slugging	5 h
Label 4 – Flow instability	15 min
Label 5 – Rapid productivity loss	12 h
Label 6 – Quick restriction in Production Choke (PCK)	15 min
Label 7 – Scaling in Production Choke (PCK)	72 h
Label 8 – Hydrate formation in production line	30 min–5 h

During production, a complex mixture of oil, gas, water, and sediments is brought to the surface, and the entire operation must be carefully managed. However, even with advanced engineering systems in place, wells remain vulnerable to undesirable production events—defined as abrupt disturbances or faults in the flow—which may interrupt operations or pose safety risks. Events such as water breakthrough into the well, unintentional closure of safety valves, flow oscillations (known as slugging), hydrate blockages, and other similar anomalies have been reported. Although such events occur infrequently, their impact is often high, leading to significant downtime, financial losses, and increased safety or environmental hazards. Therefore, the timely

identification and handling of these anomalies have been considered essential for the safe and efficient operation of offshore wells.

Undesirable events in offshore oil production can lead to serious financial and safety consequences. One extreme case is the Macondo well accident in 2010, where a failure in safety systems caused an explosion that killed 11 workers and resulted in massive environmental damage. This disaster shows how costly it can be when faults go undetected. Even regular equipment failures can be expensive—for example, using a maritime probe to fix a production issue can cost more than $500,000 per day (Andreolli, 2016). In 2016, Petrobras alone lost over 1.5 million barrels of oil, worth around $75 million, due to undetected issues in offshore wells. These examples highlight why it's so important to have smarter monitoring systems. By using machine learning and time series data, new AEM (Abnormal Event Management) systems can catch problems earlier and help reduce both risk and cost (CSB, 2014), (CSB, 2016), (Vargas et al., 2019).

To address this challenge, predictive modeling techniques have increasingly been applied. These methods have been used to analyze real-time sensor data collected from oil wells in order to anticipate undesirable events before they occur, thus enabling proactive actions to be taken—such as adjusting valves, applying chemical treatments, or shutting in the well—to prevent escalation. In recent years, the deployment of advanced sensing technologies (e.g., downhole pressure and temperature gauges, choke valve sensors) has resulted in the collection of large volumes of operational data. However, interpreting these multivariate time-series datasets and distinguishing between normal operational variations and early signs of faults has not been straightforward.

Machine learning (ML), which belongs to the broader field of artificial intelligence, has been introduced as a promising solution to this problem. It allows models to be trained on historical

data that include both normal and faulty operations, enabling automated recognition of patterns associated with specific event types. Unlike conventional physics-based models, ML approaches are capable of learning complex, nonlinear relationships without the need for explicit physical equations to describe each fault mechanism.

Despite the potential benefits, several challenges have been encountered, which have served as the motivation for this research. First, the imbalance and limited number of real event examples have made model training difficult—particularly for rare but critical cases such as hydrate plug formation. To compensate, many datasets have been supplemented with simulated or manually constructed examples provided by experts. However, care must be taken to ensure that models trained on such mixed data generalize well and do not simply memorize synthetic patterns. Second, offshore sensor data often contains noise, missing values, and gradual shifts in baseline measurements over time. Therefore, preprocessing techniques such as filtering, normalization, and imputation have been considered necessary before model training. Third, in many cases, there is no consistent agreement among domain experts regarding the precise definitions or early indicators of each anomaly type. This ambiguity has required that the structure of the modeling problem—such as how labels are assigned and predictions are made—be aligned with practical field needs and engineering judgment.

1.2 Problem statement and motivation

Given multivariate time-series data collected from offshore oil wells, the goal of this research has been defined as the development of a machine learning-based system capable of predicting undesirable production events either before or as they begin to occur. The problem has been framed as a supervised time-series classification task, where the state of the well is to be identified at each moment in time—or across short time windows—as either normal operation or

one of several defined anomaly types (such as slugging, hydrate formation, valve closure, and others). Emphasis has been placed on early detection, with the ideal outcome being the identification of an event during its transient phase—before it transitions into a fully developed steady-state fault condition.

Several challenges have been associated with achieving this objective. One major difficulty lies in the highly imbalanced nature of the dataset, where the majority of recorded data represent normal operational conditions, and only a small fraction correspond to documented anomalies. Furthermore, overlapping sensor signatures between certain types of events have made class separation more complex, and the system must be designed in a way that minimizes false alarms while still capturing true anomaly occurrences. The key question addressed in this thesis has been how to choose, train, and optimize machine learning models that can accurately learn from historical offshore well data, recognize rare event patterns, and generalize well enough to make future predictions. The broader aim is to support offshore production operations by enabling more informed and timely decision-making based on these predictive insights.

The importance of proactive anomaly detection in this context has been driven by both economic and safety considerations. A single undetected event—such as hydrate formation or severe slugging—can result in the shutdown of a well for several days, leading to substantial financial losses due to deferred production and possible equipment damage. Similarly, the unexpected closure of a downhole safety valve, although designed as a protective measure, can cause production delays and require complex interventions for reactivation. By detecting such events early, their impact may be reduced, and maintenance costs can be minimized.

Additional motivation has come from the growing industry focus on environmental safety and operational reliability. Regulatory requirements have become stricter, with increasing pressure

to avoid spills, flaring, and unplanned shutdowns. Intelligent monitoring systems have been recognized as essential tools for meeting these expectations. From a technological standpoint, the availability of the newly released "3W" dataset—which includes detailed well sensor readings and labeled event types—has provided a unique opportunity for research. As one of the first realistic, publicly accessible datasets of its kind, it has made academic work in this area possible, overcoming the previous challenge of restricted access to operational data due to confidentiality.

This research has also been aligned with the broader industrial movement towards automation and predictive maintenance, often referred to as "Oilfield 4.0." By developing, testing, and evaluating predictive modeling frameworks using real-world well data, this work has aimed to contribute to the future deployment of intelligent supervisory systems in offshore platforms. Such systems could assist operators or potentially automate certain aspects of control and maintenance, leading to safer and more efficient oil production.

1.3 Research Objectives

In response to the stated problem and motivation, this thesis sets out the following objectives:

1. **To comprehensively review offshore production challenges and the 3W dataset:** Document the types of undesirable events that occur in naturally flowing oil wells and understand their characteristics and impact. Summarize the data source (sensors, parameters, and data acquisition process) and structure of the public dataset that will be used for model development.
2. **To develop and evaluate supervised classification models for event detection:** Implement a range of established ML algorithms – including KNN, decision tree, random

forest, SVM, and artificial neural network – and train them to classify well states using historical sensor data. Optimize these models (e.g., via hyperparameter tuning and cross-validation) and assess their performance (accuracy, precision, recall, F1-score) in distinguishing normal operation from each type of anomaly.

3. **To investigate time-series modeling techniques that incorporate temporal patterns for predicting events:**

    Explore approaches such as sequence classification and forecasting models—specifically recurrent neural networks (RNNs) including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU)—to capture the dynamic evolution of sensor readings leading up to failures. These architectures were implemented to incorporate temporal dependencies into the modeling framework, addressing the limitations of static classifiers.

    Additionally, both multiclass classification and regression-based forecasting frameworks were tested. In the regression setup, models were trained to predict future class transitions using past class labels and sensor data. This formulation treated normal, transient, and anomaly states as sequential phases and aimed to forecast the upcoming class. This allowed the investigation of whether a forward-looking, continuous prediction strategy could improve early warning capability compared to fixed-point classification.

4. **To compare the performance and practical trade-offs of the different modeling techniques:**

    Analyze results to identify which methods (or combinations) are most effective for this domain, considering criteria such as prediction lead time, overall accuracy, false alarm

rate, computational complexity, and ease of deployment. Special attention was given to the impact of class imbalance, which was addressed using strategies such as class weighting and data restructuring. Performance was also evaluated across both individual well data and combined datasets to assess generalization.

5. **To provide recommendations for implementing a machine learning-based predictive monitoring system in offshore wells:**

   Based on the findings, outline best practices for data preparation, model selection, and integration of predictive analytics into existing offshore operations. Highlight any limitations of the study and suggest future work (e.g. additional data needed, potential for online learning or unsupervised approaches) to further enhance reliability. The importance of simulated data distortion was also considered, where autoencoder-based methods were used to introduce realistic variation in simulated signals. This preprocessing step was tested to determine its role in improving model robustness before deployment.

Through these objectives, the thesis bridges a gap between theoretical machine learning techniques and their application in a critical real-world industry. The next chapters detail the groundwork for achieving these goals, beginning with a survey of relevant literature on offshore production challenges, available datasets, and machine learning methods.

Chapter 2 : LITERATURE REVIEW

This chapter reviews previous work related to monitoring offshore oil production and predicting events that may disrupt operations. It begins by examining common production issues in offshore wells, such as flow instabilities, blockages, and equipment-related problems, and how

these have been studied in past research. The goal is to understand how such events affect production and what data is available to help detect them early.

The chapter also discusses how data-driven methods have been used to recognize patterns that may signal the beginning of a problem. These approaches rely on learning from historical data to find differences between normal and abnormal well behavior. Research on using time-based data is also explored, especially methods that consider how well conditions change over time. These time-based approaches are useful in cases where warning signs appear gradually before an event occurs.

In addition, studies that deal with challenges like limited data, uneven distribution of event types, and noisy measurements are reviewed. Methods to handle these challenges—such as improving data quality and preparing simulated examples—are briefly covered. Finally, this chapter includes a review of publicly available datasets, focusing on their relevance for developing and testing prediction systems in real-world offshore environments.

2.1 Production Challenges in Offshore Oil Wells

Offshore wells that flow naturally (without active pumping) can experience a variety of flow assurance and equipment-related problems during their production life. The 3W dataset considered in this study encompasses eight principal types of undesirable events identified by Petrobras experts in offshore fields. Each event represents a distinct production challenge with specific causes and effects. Below we review each of these events as documented in the literature:

1. Abrupt Increase of BSW (Basic Sediment and Water): BSW is the fraction of water and sediment in the produced fluids, typically expected to rise gradually over the well's life. An abrupt spike in BSW, however, indicates a sudden influx of water/sediment – for

instance, breakthrough from an aquifer or injection water. This surge can lead to problems such as flow assurance issues, reduced oil production, and increased corrosion/incrustation in equipment. (Vargas et al., 2019) note that automatic detection of a sudden BSW increase would allow operators to take mitigating actions (like adjusting production rates or water injection) to avoid these problems. In essence, a sharp BSW jump is a warning sign that the well's fluid composition has changed unfavorably, potentially clogging separators or pipelines if not addressed.

2. Spurious Closure of DHSV: The Downhole Safety Valve (DHSV) is a fail-safe valve installed in the well's tubing, designed to shut the well in during emergencies (e.g. if the wellhead is damaged or a platform emergency). A spurious closure refers to the DHSV accidentally closing without a legitimate trigger. Such false trips can happen without clear surface indications (the hydraulic control line might lose pressure subtly). A spurious DHSV closure halts production and incurs downtime while the valve is reopened or reset. Early identification of this event by monitoring downhole pressure signatures could enable swift corrective action (e.g. cycling the valve open) to minimize production loss and operational costs. In practice, distinguishing a spurious closure from normal operation or other faults is challenging, but it is a critical event to catch because the well is essentially sealed off when it occurs.
3. Severe Slugging: Severe slugging is an extreme form of flow instability characterized by large periodic oscillations in flow rate and pressure. Classic severe slugging in risers or flowlines involves alternating liquid build-up and gas surges, often with periods on the order of 30–60 minutes. This results in cyclic stress on pipelines and separators due to the high amplitude pressure spikes and flow changes. Severe slugging can even damage

equipment if unchecked. Key signs of slugging are regular cycles of pressure rising (as liquid accumulates) followed by sudden drops when the slug clears. Because the oscillation frequency is well-defined and relatively low, sensors along the production line (from downhole to the platform) can detect this pattern. Detecting severe slugging in advance allows operators to adjust choke settings or gas lift injection to break the slugging cycle. The literature emphasizes that early slugging detection is crucial to protect equipment and maintain steady production.

4. Flow Instability: Flow instability is a milder, less periodic oscillation in the well's flow, distinguished from severe slugging by its irregular timing. During flow instability, one or more monitored variables (pressure, temperature, flow rate) exhibit noticeable fluctuations of moderate amplitude. Unlike severe slugging, there is *no fixed period* to these oscillations – they occur in a chaotic or random manner. This lack of periodicity is a key identifier that (Theyab, 2018) helps separate instability from slugging events. Flow instabilities are important to catch because they can be a precursor to severe slugging. If an instability event is detected, it gives engineers a chance to intervene and stabilize the flow (for example, by choking back the well slightly) before it escalates into a full slugging scenario with more serious consequences. Thus, an ML model that can recognize the signatures of flow instability (e.g. simultaneous small oscillations in wellhead pressure and temperature) would provide a valuable early warning tool.
5. Rapid Productivity Loss: The productivity of a well refers to its ability to produce fluids given a certain reservoir pressure and flow conditions. Rapid productivity loss means the well's output drops sharply in a short time, indicating a sudden decline in reservoir drive or well deliverability. Several factors can cause this: e.g. an abrupt increase in water cut

(as in BSW event), plugging of the reservoir perforations, or other reservoir changes. When the reservoir energy (pressure) can no longer overcome the flow resistance, the well might slow to a trickle or stop flowing. Early identification is key – if operators realize the well is heading towards ceasing flow, they can adjust the operating conditions (such as reducing back-pressure via choke or activating artificial lift if available) to sustain production. Automatically detecting a rapid productivity loss event could rely on recognizing patterns like a simultaneous drop in downhole pressure and flow rate. This event can overlap with other phenomena (it may be the result of other events, like scaling or water breakthrough), so context is important. Nonetheless, it is treated as a distinct class in the dataset, defined by the outcome (production rate collapse) regardless of cause.

6. Quick Restriction in PCK: The Production Choke (PCK) is a surface valve that regulates the well's flow at the platform. A "quick restriction" in the PCK means the choke valve has been suddenly and significantly closed (narrowed) in a short time, more than what normal operations require. This term comes from internal Petrobras jargon – essentially if the choke opening is reduced by more than a certain threshold (e.g. >5%) in a very short interval (<10 seconds), it qualifies as a quick restriction. Such an event is often inadvertent, perhaps due to an operator error or a control system issue, and it can cause an abrupt pressure increase upstream and disruption in flow. From a data perspective, a quick choke restriction is visible as a step-change in the choke position signal (if available) or inferred from pressure jumps. Detecting this automatically is useful because the sooner it's recognized, the sooner the choke can be re-adjusted to normal, thus preventing an extended production loss or pressure surge. It is worth noting that not all wells have high-frequency

choke position data; in the dataset, this event's signature may appear indirectly via pressure trends.

7. Scaling in PCK: Scaling refers to the accumulation of mineral deposits (such as calcium carbonate or barium sulfate) inside the production choke. Over time, scale can partially clog the choke, reducing the effective flow area and thereby cutting down production rates. This tends to happen gradually, but once scale buildup is significant, the flow restriction behaves like a choke partially closed (Figure 2-1). The importance of monitoring the choke for scaling is high because scale can "dramatically reduce oil and gas production" if left unchecked. In practice, scale buildup might be detected by a slowly increasing pressure drop across the choke for a given choke setting. Automatic identification of scaling allows timely remediation (for example, injecting scale inhibitor chemicals or scheduling a clean-out). (Vargas et al., 2019) highlight that detecting this condition early can avoid long-term losses by enabling interventions before the well's flow is severely restricted. In the dataset, scaling events may not have a dramatic "event" signature like other anomalies; instead, they might appear as a prolonged deviation from normal behavior, which can be tricky for standard algorithms to catch without specialized trend analysis or time-window features.

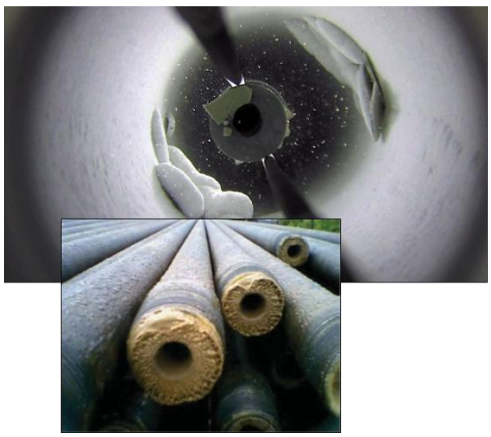

Figure 2-1. Scaling in PCK (taken from www.zerlux.com/hard-scale-removal)

8. Hydrate in Production Line: Gas hydrates are ice-like crystalline solids formed by water and natural gas at high pressure and low temperature. (Figure 2-2)

Figure 2-2. Hydrate formation in natural gas transmission pipeline (Zarinabadi & Samimi, 2012)

Offshore production lines, especially in deepwater environments, are prone to hydrate formation if fluids are not properly managed (e.g., insufficient methanol injection or insulation). A hydrate plug can completely block the flow in a pipeline or well. Hydrate formation is one of the most dreaded flow assurance problems because it can halt production for days or weeks and requires complex mitigation (like depressurization or heating) to dissociate the hydrate. The dataset's hydrate events refer to instances where hydrate likely formed in the production line, evidenced by a sudden pressure build-up and then decline as the well dies. Avoiding hydrates is so critical that operators try to predict conditions leading to hydrates and take action beforehand. Automatic detection of early signs (for example, unusual cooling trends or pressure drops indicative of a forming hydrate) could trigger preventative measures (like increasing inhibitor injection or warming the flowline). Literature notes that hydrates generally occur only in wells where the right combination of gas, water, pressure, and temperature exist – often in gas

producers, but oil wells can form hydrates as well, especially after shutdowns. The 3W dataset includes a few hydrate cases, underscoring how critical and rare this event is.

These eight events cover a range of production challenges, from fluid composition changes (BSW) and reservoir deliverability issues (productivity loss) to mechanical faults (valve closures) and flow assurance problems (slugging, hydrate, scaling). Table 2.1 summarizes the frequency of these events in the public dataset. Each event is assigned a class label (1–8, with 0 for normal operation) and is represented by multiple recorded instances in the data. Some events (like flow instability) have hundreds of real occurrences logged, whereas others (like hydrate or scaling) are extremely scarce, with only a few real examples – supplemented by simulations to aid modeling.

Table 2-1. ***Number of data instances for each event type in the 3W dataset (Vargas et al., 2019)***

Event (Class)	Real Instances	Simulated	Hand-Drawn	Total Instances
0 – Normal operation	597	–	–	597
1 – Abrupt increase of BSW	5	114	10	129
2 – Spurious closure of DHSV	22	16	0	38
3 – Severe slugging	32	74	0	106
4 – Flow instability	344	0	0	344
5 – Rapid productivity loss	12	439	0	451
6 – Quick restriction in PCK	6	215	0	221
7 – Scaling in PCK	4	0	10	14
8 – Hydrate in the production line	3	81	0	84

As shown above, the dataset contains a total of 1,984 instances (time-series sequences), of which about half are "normal" operation segments and the rest are anomaly events of various types. Notably, many events have more simulated instances than real – for example, class 5 (rapid productivity loss) has 12 real vs 439 simulated – reflecting efforts by the dataset creators to augment rare phenomena with realistic simulations. Each instance in the data is a contiguous time series that typically starts in a normal state, goes through a transient phase, and ends in a steady anomalous state. The length of each instance varies but often covers hours of data around the event. This structure is important because it means a classification model can potentially learn not just static differences between normal and abnormal but also the temporal evolution of a fault. In practice, however, many studies simplify the task to distinguishing entire sequences or specific windows labeled as "event" vs "no event".

In summary, offshore production systems must contend with a diverse set of failure modes. The reviewed events highlight why a one-size-fits-all detection approach is difficult: Some anomalies are characterized by *gradual trends* (scaling, productivity loss), others by *abrupt changes* (valve closures, quick choke changes), and others by *oscillatory patterns* (slugging, instability). This diversity necessitates a flexible and data-driven approach to event prediction – motivating the use of machine learning on comprehensive datasets like 3W, which encapsulate all these challenges in a unified framework.

2.2 3W Dataset: Data Source, Sensors, and Structure

The analyses presented in this thesis are based on the 3W dataset, which was introduced by (Vargas et al., 2019) as "a realistic and public dataset with rare undesirable real events in oil wells." The dataset was developed through a multi-year project in which examples of various

production anomalies were collected and simulated to support research on automated abnormal event detection and management. In this section, key aspects of the dataset are summarized, including its origin, the types of sensors and variables included, and the structure in which the data is organized.

2.2.1 Data Source and Collection

The 3W dataset was developed in collaboration with Petróleo Brasileiro S.A. (Petrobras), and was based on their historical well monitoring data. Significant effort was invested by petroleum engineers to validate and label past occurrences of undesirable events for inclusion in the dataset. Since truly real recorded instances of some events were either scarce or unavailable in certain wells, the dataset was supplemented with two additional sources: simulated and hand-drawn events. Simulated events were created using process simulators or realistic models to replicate the behavior of sensor readings during specific anomalies. Hand-drawn events were produced by experts who outlined expected sensor trends in scenarios where simulations were not feasible, to serve as reference patterns. These supplementary data sources were included to ensure that a sufficient number of examples were available for training algorithms, as collecting real events would have required years of observation. Overall, the dataset was composed of instances from three sources—real, simulated, and hand-drawn—which led to its naming as "3W" (referring to the three sources of well events). To enrich the dataset and reduce class imbalance, Petrobras also generated simulated instances of undesirable events using OLGA, a dynamic multiphase flow simulator developed by Schlumberger (Schlumberger, 2014). OLGA is widely used in the oil and gas industry for modeling transient flow behavior in wells and pipelines. OLGA was selected because it is Petrobras' standard tool for scenario simulation and one of the few capable of accurately modeling dynamic transients in offshore oil production systems (Vargas et al., 2019).

The data was made publicly available through an online repository—first provided as a supplementary link by Elsevier and later hosted on the UCI Machine Learning Repository—so that it could be accessed by researchers and practitioners. This openness was considered a major step forward, as oil well event data had previously been kept proprietary. Since then, the 3W dataset has been used in several studies where different detection algorithms have been explored.

2.2.2 Sensor Variables and Monitoring Setup

Figure 2-2 provides a schematic overview of a typical offshore oil well configuration as represented in the 3W dataset. The diagram illustrates the main components of a naturally flowing subsea well, including the production tubing, subsea Christmas tree, downhole safety valve (DHSV), permanent downhole gauge (PDG), and temperature-pressure transducer (TPT) located at the wellhead. These components correspond to the sensor locations from which the eight process variables were recorded. The figure helps to contextualize the placement and function of the sensors used in data acquisition and supports the understanding of the spatial distribution of pressure and temperature readings throughout the well system. This layout reflects the standard of the production setup described in the dataset and reinforces the operational basis for variable selection.

Each data instance in the 3W dataset is represented as a multivariate time series recorded from a single offshore oil well. The dataset is centered on naturally flowing offshore wells that are typically instrumented at key points along the wellbore and production line. As stated by (Vargas et al., 2019), the selection of variables was determined based on commonly available instrumentation in Petrobras operations, as well as the operational reliability of those instruments under offshore environmental conditions.

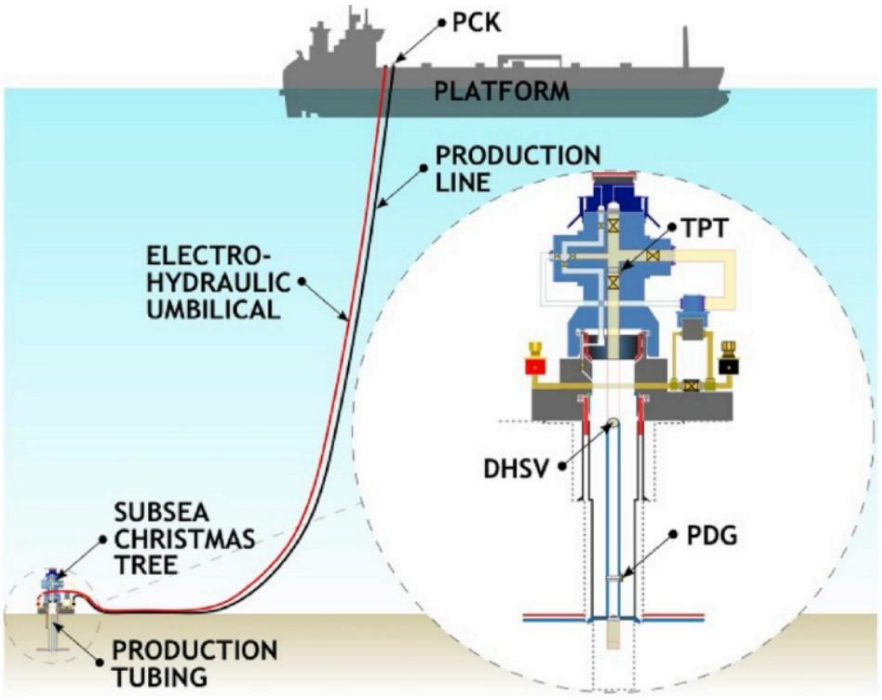

Figure 2-3. Simplified schematic of a typical offshore naturally following well. (Vargas et al., 2019)

Eight process variables were recorded for each well over time:

- P-PDG: Pressure at the Permanent Downhole Gauge. The PDG is a sensor placed in the well's tubing, often near the reservoir (downhole), providing bottom-hole pressure data.
- P-TPT: Pressure at the Temperature and Pressure Transducer on the subsea Christmas tree. The TPT is an assembly that measures both pressure and temperature at the wellhead (on the seabed).
- T-TPT: Temperature at the TPT (tree). This is essentially the wellhead fluid temperature reading.

- P-MON-CKP: Pressure upstream of the Production Choke (often called monitoring pressure). This sensor is on the platform side of the choke, measuring the tubing pressure just before the fluid passes through the choke valve.
- T-JUS-CKP: Temperature downstream of the Production Choke ("jus" meaning just downstream). It measures fluid temperature right after choking, which can drop due to the Joule- Thomson effect. (Cengel & Boles, 2002)
- P-JUS-CKGL: Pressure downstream of the Gas-Lift choke (if gas lift is present). Although the wells are naturally flowing, many are equipped with gas lift valves as a backup. This sensor would measure pressure after any gas injection point. In pure natural flow operation, this might be static or not used, but it's included for completeness.
- T-JUS-CKGL: Temperature downstream of the gas-lift choke. Similarly, this monitors the temperature at the gas lift injection point downstream.
- QGL: Gas lift flow rate, in case gas lift is active (in natural flowing mode this might be zero).

All variables were recorded with standard engineering units: pressure in Pascal's (Pa), temperature in degrees Celsius (°C), and flow rate in cubic meters per second (m^3/s). In addition to these sensor readings, each row in the dataset contains a timestamp and a class label indicating the operational status or anomaly type occurring at that moment. The data was sampled at one-second intervals for each variable—an interval deemed sufficient to capture the operational dynamics, since most events develop over several minutes or longer.

It should be noted that changes are not uniformly present across all variables during each event. Some sensors respond more directly to specific types of events. For instance, a DHSV (Downhole Safety Valve) closure typically results in a rapid drop in P-PDG, whereas T-TPT may

remain unaffected initially. In contrast, hydrate formation may first manifest as a cooling trend in T-JUS-CKP and a subsequent pressure increase in P-MON-CKP. The availability of multiple sensors ensures that a broader view of the system's behavior is provided, offering diverse input patterns from which machine learning algorithms can learn.

2.2.3 Structure and Format

The dataset is organized as a collection of CSV (comma-separated value) files. Each file corresponds to one instance (one occurrence of an event type, or a segment of normal operation). The files are grouped in folders by event type, so all instances of class 1 (BSW increase) are in one folder, class 2 (DHSV closure) in another, and so on. The naming convention of files encodes the source (real, sim, or hand) and a sequence number, and in some cases a timestamp for real events. For real event instances, the data usually encompasses three phases: a period of normal operation, then a transient period where the event unfolds, and finally a steady-state anomaly condition. Table 2.1 already indicated how many points fall into each phase for the real events (for example, a hydrate event instance might include a few minutes of normal readings, then a transient spike over some tens of minutes, and then a stabilized blocked state). This segmentation is useful for analysis – in fact, some studies treat the problem as detecting the transient or predicting the steady-state onset.

The creators also defined some benchmark tasks for using the dataset. One recommended task is a one-class classification approach: train on normal data and attempt to detect any abnormal event as an outlier. Another is a multi-class classification of each time sample or short window into one of the 0–8 classes, under certain constraints to avoid data leakage (ensuring that training and testing samples come from different instances). These task definitions highlight the challenges of using 3W: the high-class imbalance and the potential for the same instance's data to appear in

both training and testing if not careful. In our literature review and experiments, we observe how different researchers handled these issues (for instance, (Carvalho, 2021) used a group k-fold approach to avoid contamination).

In summary, the 3W dataset provides a rich, multi-sensor view of offshore well behavior for both normal and anomalous conditions. Its combination of real and synthetic data, while necessary, adds complexity in ensuring models trained on it will generalize. Nevertheless, it serves as an excellent testbed for developing and benchmarking machine learning models aimed at predicting rare events in oil production. Next, we review the machine learning methods that have been or could be applied to such data, covering both classical supervised classifiers and specialized time-series models.

2.3 Machine Learning Methods for Event Prediction

Machine learning is understood as a broad set of algorithms through which computers are enabled to identify patterns and make predictions based on data. In the context of offshore well event prediction, the focus is placed on supervised learning, where algorithms are trained on labeled examples of different well states (such as normal operation or specific anomaly types) and are then used to classify new data into the appropriate categories. Time-series modeling techniques are also considered, as they are particularly well-suited for sequential sensor data. In this section, the main types of supervised models that have been used in previous studies are reviewed—this includes those that have been implemented in this thesis, such as k-Nearest Neighbors, Decision Trees, Random Forests, and Artificial Neural Networks. Additional models that have shown potential, such as Support Vector Machines and ensemble boosting methods, are also discussed. Finally, several time-series modeling approaches, including deep sequence models, are examined for their ability to capture the temporal nature of offshore oil well anomalies.

2.3.1 Supervised Classification Algorithms

K-Nearest Neighbors (KNN): KNN is one of the simplest classification techniques based on the idea of similarity in feature space. To classify a new data point, KNN looks at the "distance" (typically Euclidean) to training data points and assigns the majority label of the closest *k* neighbors. This method is instance-based (non-parametric), meaning it stores the training dataset and defers generalization until a query is made. KNN has been successfully used in many domains due to its simplicity and relatively strong performance on well-separated classes. In the oil production context, KNN could be employed on statistical features extracted from time windows of sensor data – as done by some researchers for fault diagnosis tasks – where it would classify an interval as normal or faulty by comparing it to known examples. One advantage of KNN is that it makes no assumption about data distribution, which is useful given the often non-linear nature of well sensor patterns. However, it can be sensitive to the choice of features and scales; proper feature normalization is usually required so that variables like pressure (with large magnitude) do not overpower others. KNN also tends to be less efficient for large datasets since all distance computations must be done at query time. In their comparative study, Cover and Hart (1967) originally demonstrated the effectiveness of the nearest neighbor rule in classification, and it remains a baseline method. (Cover & Hart, 1967). In this case, KNN serves as a straightforward benchmark to see how a non-parametric approach fares on the 3W dataset events.

Decision Tree (DT): Decision trees are a family of rule-based classifiers that partition the feature space into rectangles by applying a series of binary decisions (threshold tests on features). The tree is built (usually via a greedy algorithm like CART or C4.5) by selecting features and split points that best separate the classes, often measured by information gain or Gini impurity

reduction. The end result is a tree where each leaf node corresponds to a class prediction, defined by a path of conditions (e.g., "if P-PDG < X and T-TPT > Y then class = 4 (flow instability)"). Decision trees are interpretable – a big plus in the oil industry, as engineers prefer models whose reasoning they can follow. For example, a tree might reveal that a certain pressure drop pattern is the key indicator for a particular event, aligning with engineering intuition. In terms of performance, a single decision tree can be prone to overfitting, especially in high-dimensional data with many noisy features. But they are very fast to evaluate and can capture non-linear feature interactions by their branching structure. Quinlan provides a seminal description of decision tree induction (the ID3 algorithm), and many improvements have been made since.(Quinlan, 1986) In prior oilfield event studies, decision trees have been used due to their simplicity and ease of deployment in real-time systems. In their study, Turan et al. (2021) applied a decision tree classifier to the 3W dataset, achieving an F1-score of 85% on test data. The classifier demonstrated high accuracy in detecting real events, with most events classified at over 90% accuracy.(Turan & Jäschke, 2021) While that result might not generalize to all evaluation methods, it underscores that a well-pruned tree can indeed be very effective for this problem, perhaps because it naturally isolates distinct ranges of sensor behavior corresponding to different events.

Random Forest (RF): Random Forests, introduced by Breiman, are an ensemble learning method that addresses the instability of single decision trees by averaging many of them. (Breiman, 2001) An RF classifier builds numerous decision trees, each on a bootstrap sample of the training data and typically using a random subset of features at each split (this randomness ensures the trees are diverse). The forest's prediction is the majority vote of its trees. RFs have become extremely popular for structured data classification due to their high accuracy, resistance to overfitting, and ability to handle large feature sets. In the context of oil well event detection, RF has shown

excellent results. Marins used a random forest model to detect faults in oil wells and flowlines, achieving robust performance; in one experiment on the 3W dataset, their RF achieved about 87% accuracy in identifying spurious DHSV closures (Marins et al., 2021). Bruno Carvalho also found RF to be the top performer in his evaluation of techniques for flow instability detection, with accuracies exceeding 99% in certain scenarios (Carvalho, 2021). These high scores, however, came with careful cross-validation to avoid bias and were likely for specific binary classification setups; in a more challenging multi-class early detection task, performance would be lower. Nonetheless, the strengths of RF are highly relevant: it can model complex non-linear boundaries (by the combination of many tree rules), it naturally handles interactions between sensor variables, and it provides an internal estimate of feature importance (helpful for understanding which sensors matter most). One caveat is that RF, being a collection of many trees, sacrifices the interpretability of a single decision tree – although one can still analyze which features were most used. Computationally, RF can train and predict quickly enough for moderate-sized data (especially with the number of features we have, which is just 8) and can be parallelized. It also deals well with missing data and outliers by its voting mechanism. Given these advantages, random forests are often a go-to model in tabular data problems and serve as a strong benchmark in this study.

Artificial Neural Networks (ANN): ANNs are a class of models inspired by the brain's neural networks, capable of approximating very complex functions. In supervised classification, a typical ANN (specifically a Multi-Layer Perceptron, MLP) consists of an input layer (taking our sensor features), one or more hidden layers of interconnected "neurons" with non-linear activation functions, and an output layer that produces class probabilities. The network's weights are trained through examples using algorithms like backpropagation (Rumelhart et al., 1986). Neural networks are known for their ability to capture non-linear relationships and interactions between

features that might be hard to model with simpler classifiers. In oil well event detection, an ANN might, for example, learn a combination of sensor thresholds and rates-of-change that signal an upcoming fault – patterns that are not easily expressed as a single rule. Early applications of ANNs in petroleum engineering (1990s and 2000s) showed success in areas like well performance prediction and seismic interpretation (Mohaghegh, 2000). With modern advances (deep learning), neural nets have become even more powerful. However, for the scope of this thesis, we focus on a relatively small MLP as one of the classifiers rather than very deep networks. Prior work by Fernandes Júnior utilized a neural network for anomaly detection in oil producers, and others have considered Extreme Learning Machines (a form of single-layer neural network) for fast learning (Fernandes Jr et al., 2024). A challenge with ANNs is the need for tuning many hyperparameters (number of layers, neurons, learning rate, etc.) and the risk of overfitting, especially with limited training data for the rare classes. In this dataset, since there are thousands of samples (time points) but effectively only dozens to hundreds of event instances, care must be taken not to train an overly large network. Another point is that ANNs treat input features as a vector without inherent temporal structure – thus, they might require handcrafted time-domain features (e.g., the slope of pressure or a time window's FFT components) unless combined with a sequence model. Despite these considerations, ANNs provide a flexible modeling approach that, with sufficient data, could potentially outperform more rigid models. The success of deep learning in other industries suggests that, as more oil production data becomes available, neural networks will play a growing role in predictive analytics.

Support Vector Machines (SVM): While not originally listed among the four primary models, SVM is a well-known supervised classifier worth discussing as an alternative. SVMs aim to find an optimal separating hyperplane between classes by maximizing the margin (distance)

between training points and the decision boundary, often after projecting data into a higher-dimensional space via a kernel trick (Cortes & Vapnik, 1995). SVMs have been successfully applied in fault diagnosis scenarios due to their ability to handle non-linear class boundaries (with kernels like RBF) and their robustness against overfitting in high dimensions (thanks to the margin maximization principle). In the context of the 3W dataset, an SVM could be trained to distinguish, say, normal vs abnormal, or even multi-class (though multi-class SVM requires combining binary classifiers). Some studies (e.g., (Marins et al., 2021)) included SVMs in their comparison; however, SVMs can be less practical when the dataset is very large (training complexity grows with the number of samples) and when multi-class with many classes is needed. For our problem with 9 classes (0–8), one-versus-one or one-versus-all SVM would lead to multiple decision functions. Nevertheless, SVM's ability to work with kernel functions could capture relationships like the "shape" of time-series if one uses a dynamic time warping kernel or other specialized kernels for time-series. One specialized use in well event detection has been one-class SVM, which is trained only on normal data to recognize the normal class, flagging anything else as an outlier; Fernandes Júnior explored such an approach for oil well anomalies (Fernandes Jr et al., 2024). In summary, SVM remains a strong candidate, especially for binary classification tasks in this domain, though we must weigh its complexity for multi-class problems.

Each of the above supervised models has strengths and weaknesses (Table 2.2). Many studies might compare them on criteria such as interpretability, training speed, and suitability for high-dimensional vs small-sample scenarios. For instance, Decision Trees are interpretable but can overfit; Random Forests improve stability but lose some interpretability; KNN is simple but may struggle with noisy or irrelevant features; SVM handles complex boundaries but is computationally heavy; Neural Networks can model intricate patterns but require lots of data and

tuning. A key insight from prior comparisons (e.g., (Caruana & Niculescu-Mizil, 2006)) is that no single algorithm uniformly dominates others across all tasks– which justifies our experimental approach of trying multiple models. In fault detection literature specifically, tree-based ensembles and SVM have often performed well, but neural networks are catching up as data availability increases. Bruno Carvalho provides a recent example: his initial tests included 1-NN, LDA, QDA, GNB, and RF, and among those, the RF was best; he suggests exploring deep learning as future work (Carvalho, 2021).

Table 2-2. Summary of Supervised Learning Models Used for Well Event Classification

Model	Strengths	Weaknesses
K-Nearest Neighbors (KNN)	Simple to implement; makes no assumptions about data distribution; effective on well-separated data.	Sensitive to feature scale and irrelevant features; slow at inference time for large datasets.
Decision Tree (DT)	Highly interpretable; fast prediction; captures non-linear feature interactions; suitable for real-time systems.	Prone to overfitting; sensitive to small variations in data.
Random Forest (RF)	High accuracy; robust to overfitting; handles feature interactions; provides feature importance; efficient.	Less interpretable than single trees, can become large in size; tuning may be required.
Artificial Neural Network (ANN)	Models complex, non-linear relationships; adaptable with sufficient data; proven success in anomaly detection.	Requires large labeled datasets; many hyperparameters to tune; risk of overfitting with small data.
Support Vector Machine (SVM)	Strong theoretical foundation; works well with high-dimensional data; effective with small-to-medium datasets.	Computationally expensive on large datasets; complex for multi-class problems; sensitive to kernel choice.

2.3.2 Time-Series Models and Sequence Learning

The supervised models described earlier generally treated data as independently and identically distributed (IID) samples, often following the extraction of features from fixed-length time windows. However, the raw sensor data encountered in this problem is inherently sequential, consisting of time-series data collected from offshore wells. Temporal dependencies—such as evolving trends and transitions leading to failure—may not be fully captured by static classifiers unless the time dimension is explicitly considered. Therefore, models that directly account for temporal patterns have been explored.

Sequence Learning with Recurrent Neural Networks (RNNs): Recurrent neural networks (RNNs) are designed to model temporal dependencies in sequential data by maintaining a hidden state that evolves with each time step. A widely adopted variant is the Long Short-Term Memory (LSTM) network (Hochreiter & Schmidhuber, 1997), which introduces memory gates to address vanishing gradient problems and retain long-term information. LSTMs have been applied in industrial domains for detecting rare faults and time-dependent anomalies in sensor signals. Gated Recurrent Units (GRUs), proposed by Cho (Cho et al., 2014) offer a simplified alternative with fewer parameters while retaining similar performance and have also been used in scenarios with limited labeled data.

Time-Series Decomposition (e.g., STL): Seasonal-Trend decomposition using Loess (STL) is a classical approach to separate a time-series into trend, seasonal, and residual components (Cleveland et al., 1990). It is particularly useful for understanding the underlying structure of sensor data, identifying abnormal patterns in the residual component, and informing

further modeling. In oil and gas applications, STL decomposition has been used to extract relevant features for fault detection models or for visual diagnostics.

Autoencoders for Anomaly Detection: Denoising autoencoders are a class of unsupervised neural networks used to learn compressed representations of input data and reconstruct clean signals. When trained on normal operational sequences, autoencoders can reconstruct known patterns but perform poorly on unseen anomalies, making them suitable for anomaly detection (Sakurada & Yairi, 2014). This technique has been applied to industrial settings, including well monitoring, where reconstructed signals are compared with input signals to identify deviations that may indicate early stages of failure.

Class-Imbalance Handling and Class Weighting: In many industrial time-series datasets, the majority of samples represent normal operation, while anomalies are rare. This class imbalance can bias learning algorithms. One widely used solution is the application of class weighting during training (He & Garcia, 2009), which penalizes the loss function more heavily for underrepresented classes. This technique has shown improvements in fault detection performance across domains, especially when combined with deep sequence models.

Regression-Based Sequence Prediction: While classification is the dominant approach in fault identification, regression-based frameworks have also been explored. These models aim to forecast future sensor values or event states and detect deviations from expected behavior. In some cases, class transition modeling has been approached using regression to predict the likelihood of shifting from normal to transient to steady-state fault, offering a more continuous view of system evolution (Malhotra et al., 2017).

Hybrid and Feature-Augmented Approaches: In practice, hybrid frameworks have been proposed, where time-series models extract temporal features or anomaly scores that are then fed into supervised classifiers (Hajirahimi & Khashei, 2019). For example, LSTM autoencoders can generate residuals that serve as features for a decision tree or an ensemble classifier. These methods attempt to combine the strengths of temporal modeling and interpretability.

These techniques represent a growing field of interest within offshore production analytics. Prior work, such as (Nikitin et al., 2022), (Marins et al., 2021), and (Aldabagh et al., 2023) demonstrates the value of applying sequence models, ensemble learning, and hybrid detection frameworks for event detection in oil production systems. The techniques discussed in this section form the basis for model selection and evaluation strategies presented in later chapters of this thesis.

Chapter 3 : Methodology

3.1 Data Preparation

The 3W Petrobras dataset used in this study contains approximately 50 million time-stamped records collected from offshore oil wells. Each record includes 10 columns: pressure and temperature sensor variables, a timestamp, and a class label indicating the operating state of the well. Three additional variables—QGL (Gas Lift Flow Rate), T-JUS-CKGL (temperature downstream of the gas lift choke), and P-JUS-CKGL (pressure downstream of the gas lift choke)—were excluded from the analysis. This exclusion was based on the observation that not all wells in the dataset operate under gas lift, making these features inconsistent across the data.

The remaining variables consist of pressure sensors (indicated by the prefix “P”) and temperature sensors (indicated by the prefix “T”), which were used for modeling and analysis.

Initial data exploration revealed physically implausible values. For example, negative pressure readings were present in some wells, which are not valid in practical scenarios. Such values were removed from the dataset. Outliers were also detected using scatter plots and pairwise visualizations. Specifically, pressure values exceeding 2×10^9 Pascals in the P-TPT variable and temperature values above 150°C in the T-JUS-CKP variable were identified as anomalies and excluded from the dataset.

Further preprocessing steps—including smoothing, decomposition, and class rebalancing—will be applied based on whether the modeling approach treats data as independently and identically distributed (IID) samples or as time-series sequences. These techniques will be introduced in the following sections, depending on the specific needs of each modeling method.

3.2 Supervised Classification

In the initial modeling stage, the dataset was treated as a collection of independently and identically distributed (IID) samples. Based on this assumption, appropriate preprocessing steps were applied before training. These included removing irrelevant features, handling outliers, and scaling sensor values.

A set of supervised classification algorithms was implemented to evaluate the classification performance on this preprocessed data. Specifically, the models tested included k-Nearest Neighbors (KNN), Decision Tree, Random Forest, Artificial Neural Network (ANN), and Support Vector Machine (SVM) classifiers. Each model was trained to classify time windows into one of

the defined well states (normal operation or specific anomaly labels) using statistical features derived from the sensor values.

3.2.1 Data Pre-Processing

During the preprocessing phase, missing values were handled by carefully removing rows without critical data. Since a large number of data points were available, not all data were used. Some classes of data were merged to simplify the model (e.g., transition stages for certain labels were combined), reducing the total number of data points to approximately 40 million.

Finally, 10,000 samples were selected for further analysis. Hypothesis testing was conducted to ensure that the sample data were representative of the entire population. As a result, it was determined with 95% confidence that the selected 10,000 data points were likely representative of the whole dataset.

After sampling, outlier removal and basic cleaning steps were applied. The data were visualized using heatmaps and pair plots to identify outliers and relationships between variables. From the Spearman correlation heatmap, it is observed that pressure variables such as P-PDG, P-TPT, and P-MON-CKP are highly correlated, and temperature variables such as T-TPT and T-JUS-CKP are correlated with each other (Figure 3.1).

Using scatter plots, extreme data points that do not physically make sense were observed in the pair plot, identifying them as outliers. For example, all data above 2×10^9 Pascal in the P-TPT feature and data points above 150°F in the T-JUS-CKP feature were removed. This helped normalize the classes, making them easier to differentiate (Figure 3-2).

To enhance model robustness and simulate realistic sensor variability, Gaussian noise was considered during data preparation. Gaussian noise, also known as normal noise, refers to random values drawn from a normal distribution with a defined mean (typically zero) and standard deviation. This type of noise is commonly used in time-series modeling to reflect real-world disturbances and uncertainty in sensor readings (Goodfellow, Bengio, & Courville, 2016). Its application can help prevent overfitting by exposing the model to slightly perturbed versions of the original data. In the context of offshore oil production, where sensor readings are subject to fluctuations due to equipment sensitivity and environmental conditions, Gaussian noise provides a simple yet effective way to simulate such imperfections.

Gaussian noise (Arslan et al., 2019) was added to introduce realistic variability to the simulated data, resulting in percentage changes ranging from 1.16% (T-TPT) to 8.04% (P-MON-CKP). Higher percentage changes were observed in pressure variables (e.g., P-PDG, P-MON-CKP) due to their larger values, making these changes acceptable compared to the lower percentage changes in temperature variables (e.g., T-TPT) (Table 3-1). Overall, the noise introduced realistic variability without overwhelming the data. For each numerical variable, the noise was generated from a Gaussian distribution with a mean of 0 and a standard deviation equal to 10% of the variable's standard deviation (1). This approach ensured that the noise was proportional to the variability of each feature, preserving the dataset's structure while introducing realistic randomness to the simulated data (Figure 3.3).

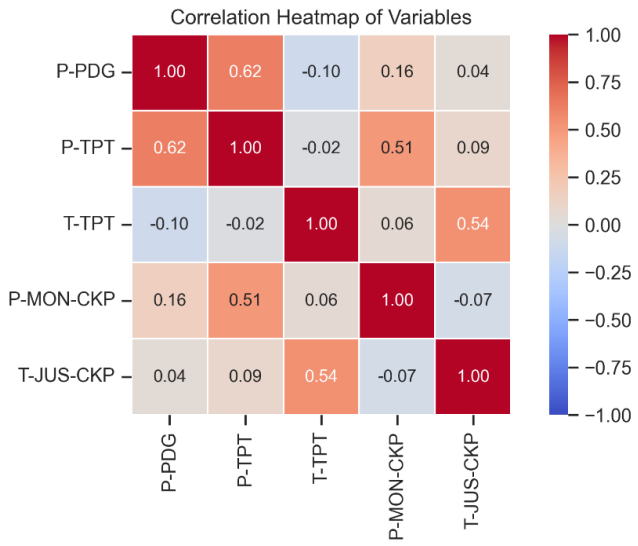

Figure 3-1. Correlation Heatmap of Variables

The formula used to add Gaussian noise to each variable is:

$$x_{noisy} = x_{original} + \epsilon \qquad (1)$$

Where x_{noisy} is the value after adding noise, $x_{original}$ represents the original value in the dataset, and $\epsilon \sim N(\mu, \sigma)$ denotes Gaussian noise sampled from a normal distribution. The noise parameters are defined as follows: $\mu = 0$, which is the mean of the noise, and, $\sigma = 0.1 \times std\_dev$, which is the standard deviation of the noise, set to 10% of the feature's standard deviation (std_dev).

Table 3-1. Variables and Their Percentage Change after Gaussian noise

Variables	P-PDG	P-TPT	T-TPT	P-MON-CKP	T-JUS-CKP
Percentage Change (%)	5.3	3.5	2.1	8.1	2.0

The GANs method can be applied in further analysis, utilizing the characteristics of real data to refine the simulated data (Feng et al., 2021).

After the data preprocessing phase, machine learning can be applied.

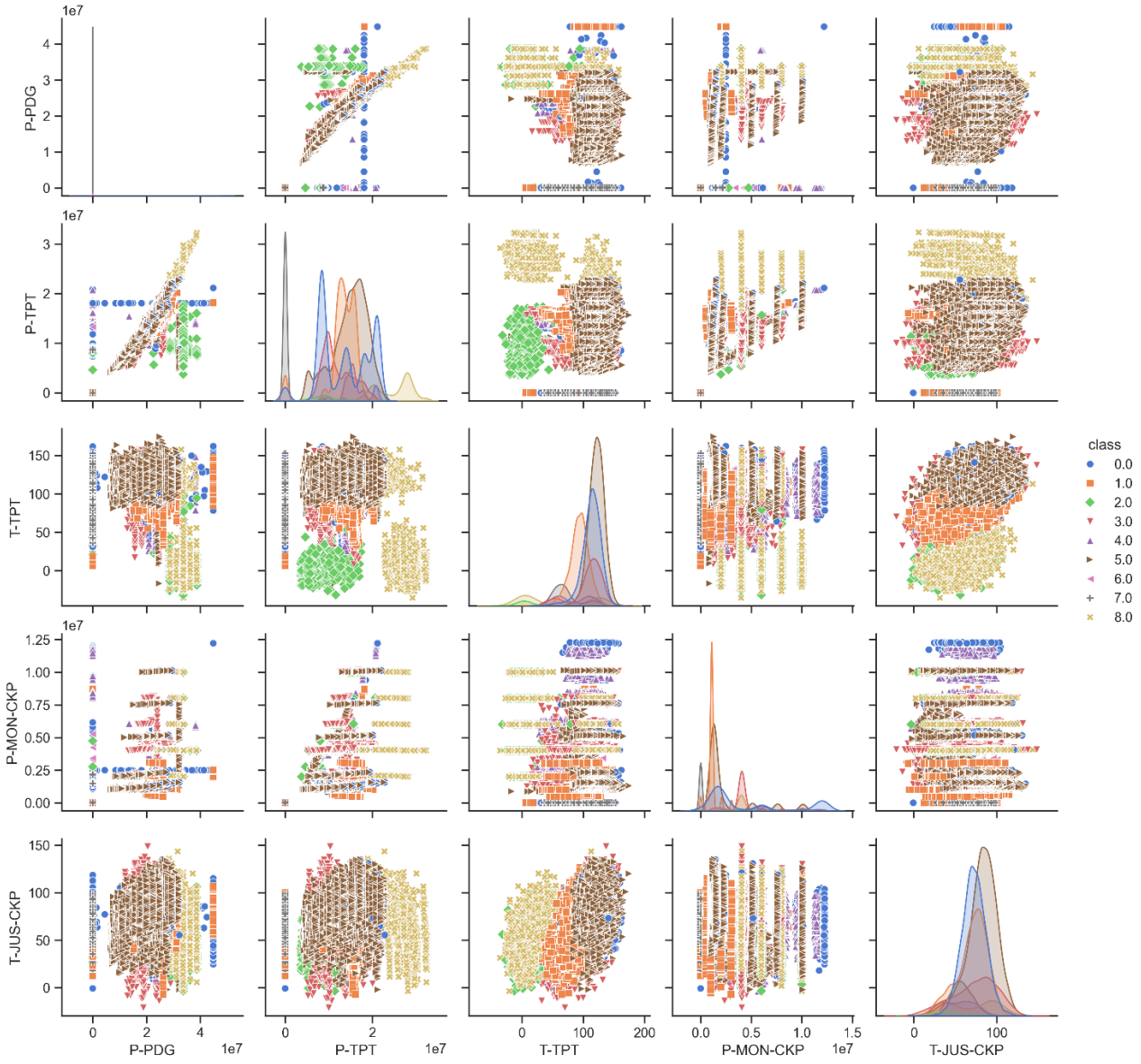

Figure 3-2. Pairplot after Outlier Removal

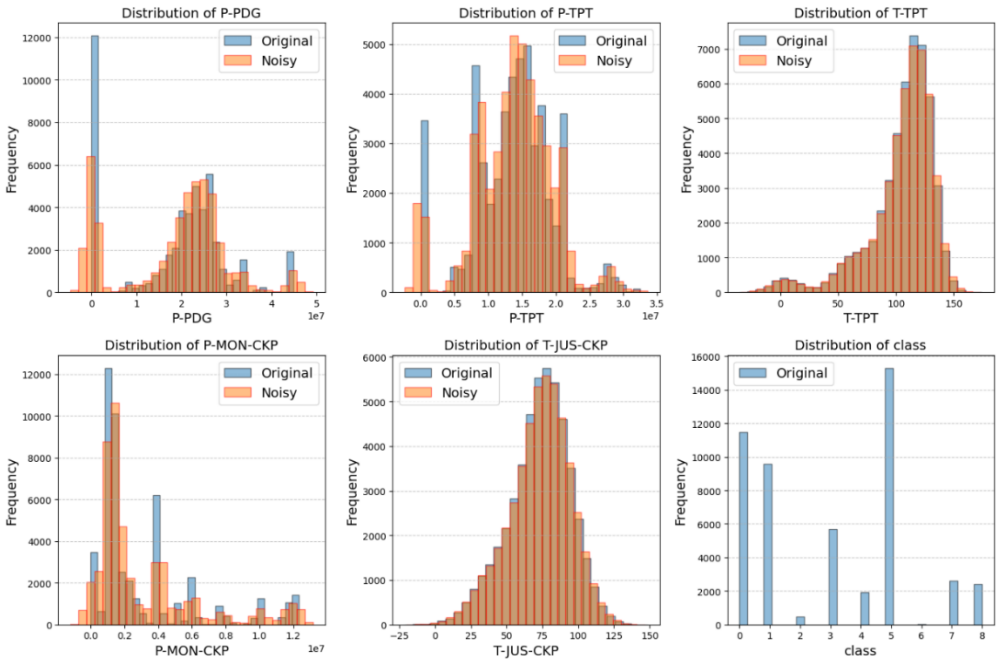

Figure 3-3. Comparison of Original and Noisy Data Distribution

3.2.2 Modeling

The problem is being addressed using a supervised machine-learning approach, as the goal is to classify data into predefined categories (undesirable events). The feature "class" which represents undesirable events in the dataset, is predicted using the following variables along with their respective importance scores: P-PDG (0.34), P-MON-CKP (0.33), P-TPT (0.19), T-TPT (0.13), and T-JUS-CKP (0.02). These importance scores indicate the relative contribution of each feature to the model's decision-making process, typically derived from how often and effectively a variable is used to split data in decision-tree-based models (Breiman, 2001). Higher scores reflect greater influence on classification outcomes, helping identify which sensor readings are most relevant for detecting abnormal conditions.

Classification models, including K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), Neural Networks (ANN), and Support Vector Machine (SVM), are being applied. These techniques were selected due to their effectiveness in handling classification

problems, their ability to work with numerical data, and their capability to identify complex patterns within the data.

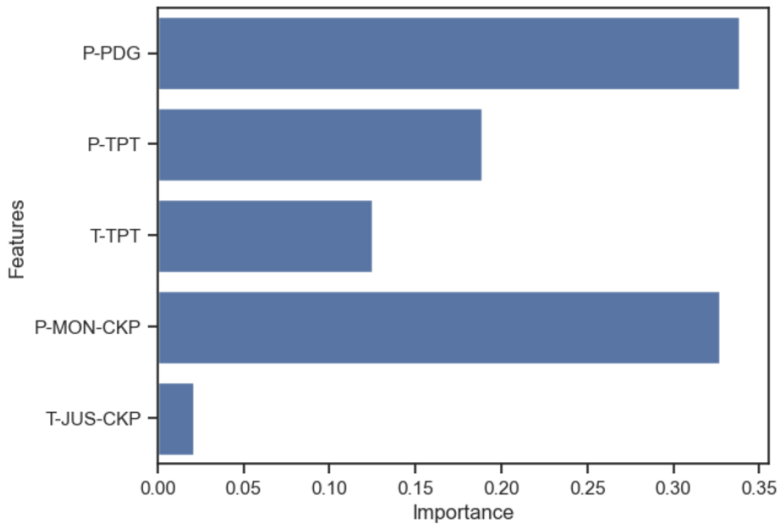

Figure 3-4. Importance scores of each variable in the 3W dataset

Supervised machine learning is appropriate for this problem because the dataset contains labeled data, making it suitable for building models that can learn to associate features with specific classes (types of undesirable events). The objective is to predict which cluster the data belongs to when an undesirable event occurs, enabling timely identification of operational issues.

In this study, 80% of the data was used for training, and 20% was used for testing. This common split strategy ensures that the model learns patterns from a majority of the data while being evaluated on a separate, unseen portion to assess generalization performance. The training set is used to fit the model, and the test set provides an unbiased evaluation of how well the model is likely to perform on new, real-world data. This method helps detect overfitting—where a model performs well on training data but poorly on new data—and ensures a more realistic estimate of predictive accuracy (Kohavi, 1995).

Due to the significant differences in feature values (e.g., pressure and temperature), feature scaling was applied to ensure that all variables contributed equally to the model. Scaling helps prevent features with larger numeric ranges from disproportionately influencing the learning process—especially important for distance-based algorithms like K-Nearest Neighbors (KNN), which rely on feature magnitude when calculating similarity. To maintain the integrity of the evaluation process, scaling was performed only on the training data, and the same transformation was then applied to the test data. This approach avoids data leakage, where information from the test set could indirectly influence the training process and lead to overly optimistic performance estimates. Proper feature scaling has been widely recommended in machine learning literature as a critical preprocessing step for reliable and fair model training (Jain et al., 2000).

For each model, hyperparameter optimization and cross-validation were performed using GridSearch and 10-fold cross-validation. GridSearch systematically explores combinations of model parameters to identify those that yield the best performance, while 10-fold cross-validation helps ensure that model evaluation is reliable and not overly dependent on a specific train-test split. In this procedure, the dataset is divided into ten parts, with each part used once as a validation set while the others are used for training. This repeated testing helps reduce variance in the evaluation metrics and provides a more generalizable measure of model performance. To assess the classification performance comprehensively, F1 score, accuracy, recall, and precision were used—metrics that are particularly important in imbalanced datasets, where accuracy alone may be misleading. In this study, no class balancing techniques (such as oversampling or undersampling) were applied during this classification approach. This decision was based on the observation that the models—particularly tree-based classifiers—were able to achieve promising weighted scores, indicating good performance even with imbalanced data. These evaluation strategies are

commonly applied in supervised machine learning workflows to balance bias and variance while avoiding overfitting (Yacouby & Axman, 2020).

The definitions and formulas below provide the fundamental metrics used to evaluate classification models (Saito & Rehmsmeier, 2015) (Table 3-2). Precision emphasizes how many predicted anomalies were actually correct, while recall measures how many true anomalies were successfully identified. F1 score, as the harmonic mean of precision and recall, balances the two and is particularly valuable when the cost of false positives and false negatives differs.

- True Positives (TP): Correctly predicted positive instances.
- True Negatives (TN): Correctly predicted negative instances.
- False Positives (FP): Incorrectly predicted positive instances.
- False Negatives (FN): Incorrectly predicted negative instances.

Table 3-2. Classification Metrics and Their Formulas

Metric	Formula
Precision	Precision = TP / (TP + FP)
Recall	Recall = TP / (TP + FN)
F1 Score	F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

All positive and negative predicted instances can be observed using a confusion matrix. Each model works differently and below is an explanation of how they function and why they were chosen.

K-Nearest Neighbor (KNN): KNN is a straightforward algorithm that classifies a data point by looking at the closest k data points in the training set. The class label that appears most

frequently among these neighbors is assigned to the query point (Altman, 1992). To measure the closeness, the Euclidean distance is used, calculated as (2):

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (2) \text{ (Cover \& Hart, 1967)}$$

The variable X represents the query data point to be classified, while Y refers to a data point in the training set. The symbol x_i denotes the i-th feature of X, and y_i represents the i-th feature of Y. The variable n indicates the total number of features in the dataset. And $d(X,Y)$ represents the Euclidean distance between X and Y.

The class label of X is assigned based on the most common class among the k-nearest neighbors. This method was chosen because it is simple to implement and serves as a good baseline model.

In the K-Nearest Neighbors (KNN) model, Min-Max scaling was applied to normalize the feature values to a common scale between 0 and 1. This step is critical because KNN relies on distance-based similarity, typically measured using Euclidean distance, which is sensitive to the magnitude of feature values. Without normalization, features with larger numerical values can dominate the distance computation, leading to biased classifications. To optimize model performance, hyperparameter tuning was conducted using GridSearchCV, a method that exhaustively searches over specified parameter values through cross-validation. In this study, 10-fold cross-validation was employed, where the training data is split into ten subsets, with nine used for training and one for validation in each iteration. This approach helps prevent overfitting and ensures the model generalizes well to unseen data (Kuhn & Johnson, 2013). The number of neighbors was tested for values [10, 100, 500], with 10 neighbors achieving the highest F1 score during cross-validation.

Decision Tree (DT): In this project, a Decision Tree Classifier was implemented to categorize the dataset into predefined classes representing undesirable events. The decision tree builds a hierarchical model by recursively splitting the dataset based on feature values to improve class purity. The splitting criterion used in this study was Gini Impurity, a common metric that quantifies the degree of class mixture at a node. Lower Gini values indicate purer splits, and the algorithm aims to minimize impurity at each decision point. The formula for Gini Impurity (3) is:

$$Gini = 1 - \sum_{i=1}^{c} p_i^2 \qquad (3) \text{ (Breiman, 2001)}$$

Where p_i is the proportion of samples belonging to class i.

DTs were chosen because they are easy to understand, interpret and handle numerical data well. A Decision Tree Classifier was also used to classify the dataset, with hyperparameter tuning performed using GridSearchCV and 10-fold cross-validation. The best parameters were a maximum depth of 15, a minimum number of samples required to split an internal node of 5, and a minimum number of samples required at a leaf node of 5, achieving the highest F1 score.

Random Forest (RF): Random Forest is an ensemble of decision trees. Unlike a single decision tree, which may be sensitive to training noise or overfit specific patterns, Random Forest builds several trees using different random subsets of both data and features (a process known as bootstrap aggregation or bagging). Each tree in the ensemble casts a "vote," and the final prediction is determined by majority voting. Mathematically, the prediction for a classification task is expressed as (4):

$$\hat{y} = mode\big(T_1(x), T_2(x), \dots, T_m(x)\big) \qquad (4) \text{ (Breiman, 2001)}$$

where $\hat{y}$ is the final predicted label, $T_i(x)$ is the output of the i^{th} tree for input x, and m is the total number of trees.

This method was selected due to its ability to reduce variance without substantially increasing bias, offering more robust predictions than individual models. In this project, the Random Forest classifier was trained on data normalized using Min-Max scaling, ensuring uniform feature contribution during tree construction. Key hyperparameters—such as the maximum tree depth (10), number of estimators (200), and maximum number of training samples per tree (20,000)—were tuned via GridSearchCV to optimize performance. The combination of ensemble averaging, randomized training, and careful hyperparameter tuning makes Random Forest an effective and stable choice for multi-class classification tasks involving sensor data with varying scales.

Artificial Neural Networks: The neural network model used in this study is a Multilayer Perceptron (MLP), a widely used type of feedforward artificial neural network (ANN) suitable for classification tasks (Gardner & Dorling, 1998). The model was implemented using MLPClassifier from Scikit-learn, employing ReLU activation and the Adam optimizer, both of which are default settings in the library (Pedregosa et al., 2011). The architecture initially tested included four hidden layers (15, 25, 45, 100 neurons), and through GridSearchCV with cross-validation, the optimal structure was determined to be two hidden layers of 25 neurons each, achieving the highest cross-validated F1-score.

MinMaxScaler was used to normalize input features, a critical step for neural networks to ensure faster convergence and balanced learning. The model was trained on 80% of the dataset and tested on the remaining 20%, with performance assessed using accuracy, precision, recall, and F1-score. The ReLU activation function is defined as (5):

$$f(x) = \begin{cases} x & if\ \ x > 0, \\ 0 & if\ x \leq 0. \end{cases} \qquad (5) \qquad \text{(Glorot et al., 2011)}$$

It introduces non-linearity and helps mitigate vanishing gradients. The Adam optimizer is an adaptive learning rate optimization algorithm that combines the advantages of RMSProp and momentum, making it well-suited for large datasets and sparse gradients. Neural networks were chosen for their capacity to model complex, non-linear relationships, making them well-suited for high-dimensional sensor data as in this study.

Support Vector Machine (SVM): In this study, a Support Vector Machine (SVM) was used as a supervised learning algorithm to classify events into predefined categories. SVM works by drawing a boundary (called a hyperplane) that separates different classes as clearly as possible (Hsu et al., 2003). The model was trained using Min-Max scaling to normalize feature values and avoid bias from variables with large ranges (e.g., pressure vs. temperature). To improve model performance, a parameter tuning process was applied using GridSearchCV with 10-fold cross-validation. The following parameters were selected:

Kernel: The RBF (Radial Basis Function) kernel was chosen. It allows the model to capture complex, non-linear relationships between features. RBF calculates the similarity between two data points based on their distance and gives higher weight to closer points.

C (Regularization): Set to 5, this parameter controls how strictly the model tries to avoid misclassifications. A higher C tries to classify every training point correctly but may risk overfitting.

Gamma: Set to 1, this defines how far the influence of one training example reaches. A small gamma means the influence is broader, while a large gamma focuses on closer points.

The mathematical formula for the RBF kernel is:

$$K(x, x') = \exp\left(-\gamma * \left|\left|x - x'\right|\right|^2\right)$$

The RBF kernel is a popular choice for non-linear classification tasks, as it measures the similarity between two data points $x, and\ x'$. In the formula $\left|\left|x - x'\right|\right|^2$ represents the squared Euclidean distance between the two points, while γ (gamma) controls the influence of a single training example. A high gamma value makes the decision boundary more sensitive to individual data points, focusing on local patterns, whereas a lower gamma value allows the model to consider more global trends. The exponential function transforms the distance into a similarity score ranging between 0 and 1. The key parameters optimized during training included the regularization parameter $C = 5$, which controls the trade-off between achieving a low training error and maintaining a smooth decision boundary, and the kernel parameter γ=1, which defines the shape of the decision boundary. These parameters were fine-tuned using GridSearchCV with 10-fold cross-validation to enhance model performance (Cortes & Vapnik, 1995).

3.3 Time-Series Modeling Approach

While the supervised classification models discussed above treat the data as independently and identically distributed (IID), the nature of offshore well sensor data is inherently sequential. Capturing temporal dependencies is crucial for early and accurate event detection. Therefore, the next section explores time-series modeling techniques—including LSTM, GRU, and regression-based transition prediction—to better utilize the sequential structure of the dataset and improve anomaly forecasting capabilities.

3.3.1 Sequential Nature of Data

The sensor readings in offshore production systems are recorded over continuous time intervals, which means they are not independent of one another. Instead, current measurements are often related to previous values due to underlying physical processes such as pressure buildup, temperature fluctuations, or flow changes (Jiang et al., 2019). As a result, treating each observation

as an isolated data point—as in typical supervised classification—ignores important time-based relationships that could improve the prediction of abnormal conditions.

Time-series modeling is designed to handle this type of data. These models account for autocorrelation, trends, and temporal shifts by considering the order of data points. For example, a drop in pressure or a steady rise in temperature may not signal a problem on its own, but their timing

3.3.2 Data Transformation and Preprocessing

In this time-series approach, the focus was placed on Problem 1, which corresponds to a specific type of steady anomaly- an abrupt Increase of BSW in the dataset. According to the class distribution, the most frequent anomaly is class 5, followed by class 0 (Normal Operation), and then class 1 (the selected anomaly). Problem 1 was chosen because it is not only the second most frequent anomaly but also the first labeled anomaly in the system, making it a meaningful and interpretable case for time-series modeling. Real datasets from Well-1, Well-2, and Well-6 contain this class label, and to ensure consistency and sufficient training data, the corresponding simulated datasets—SIMULATED_00001, SIMULATED_00002, and SIMULATED_00006—were also selected.

This problem was selected exclusively due to computational limitations. Even during the earlier supervised classification stage, handling 10,000 samples across all classes required significant processing time for each model and preprocessing step. Narrowing the focus to a single anomaly class allowed for more manageable model development and deeper analysis. Furthermore, Problem 1 includes a clearly defined transition stage (labeled as class 101), which makes it particularly relevant for time-series tasks such as classification and regression-based prediction. Accurately identifying and forecasting this transitional phase is critical, as it represents

a potential early warning signal before an undesirable event occurs. Therefore, this problem formulation not only aligns with the dataset structure but also supports the goal of proactively detecting operational anomalies through sequential modeling.

To prepare the data for time-series modeling, extensive preprocessing was conducted using R. This involved the following key steps:

Datetime Parsing and Class Labeling: The date and time columns were combined into a unified datetime object.

Class-Shaded Visualizations: To better understand the temporal behavior of sensor variables during different operational states, class-shaded time-series plots were generated for both real and simulated datasets. These visualizations overlay each sensor signal with background shading corresponding to the labeled class—Normal, Transient Anomaly, and Steady Anomaly—allowing clear interpretation of transitions and dynamic changes across time. A comparative analysis of the real data from WELL-00001 (Figure 3-5) and its simulated counterpart SIMULATED_00001 (Figure 3-6) revealed several key differences. In the real dataset, transitions into anomalies appeared smoother and more gradual, particularly for variables such as P.MON.CKP and P.TPT, where steady declines or increases occurred over extended periods. In contrast, the simulated data exhibited more abrupt fluctuations, with variables like P.MON.CKP showing high-frequency oscillations during the steady anomaly phase, possibly indicating overamplified behavior. While the Normal class periods in the real data showed stable trends with minimal noise, early deviations and subtle instabilities were observed in the simulated signals, suggesting premature transitions or sensitivity in the simulation process. Furthermore, the P.PDG variable remained nearly constant across all classes in the real data, whereas it followed a rising and slightly volatile trend in the simulated version, pointing to a mismatch in modeled pressure

dynamics. These differences highlight the importance of validating simulated behaviors against real observations before proceeding to model training, as discrepancies in trend smoothness and anomaly representation may impact model performance and generalization.

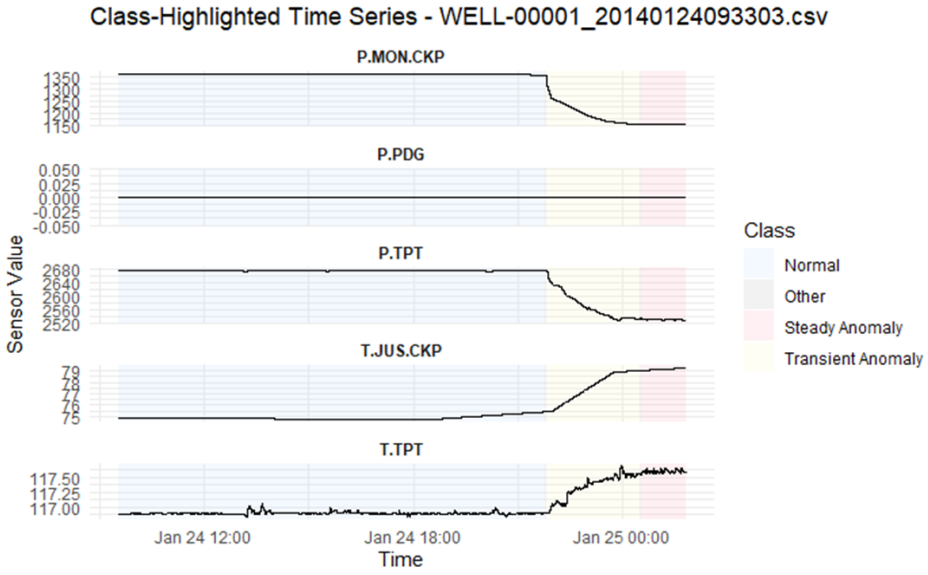

Figure 3-5. Sensor Trends with Class Overlays – Real Dataset (WELL-00001)

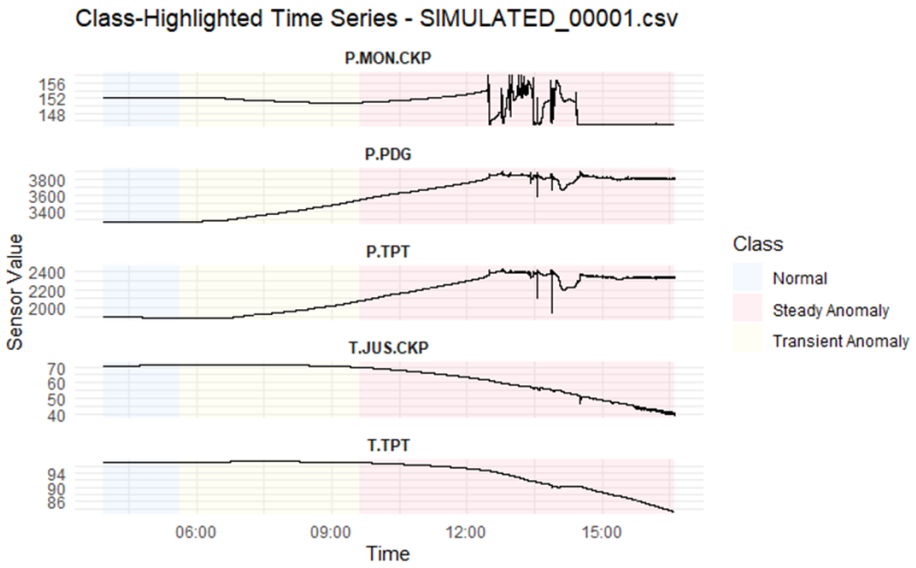

Figure 3-6. Sensor Trends with Class Overlays -Simulated Dataset (SIMULATED_00001)

Histograms and Boxplots: To assess the quality and variability of the sensor signals, distributional analysis was conducted using both histograms and class-separated boxplots. In the

real dataset (WELL-00006), variables such as T.TPT and P.TPT exhibited tight clustering with clear modal behavior, while others like P.PDG remained nearly constant, indicating stable operational conditions. In contrast, the simulated dataset (SIMULATED_00006) showed exaggerated variability, multimodal distributions, and frequent artificial peaks—particularly evident in variables like P.MON.CKP and T.JUS.CKP. These discrepancies highlighted the need for additional preprocessing steps to harmonize data quality across sources. The boxplots further revealed class-specific behaviors and outliers concentrated in transition and anomaly phases, providing valuable cues for model sensitivity to class shifts. These findings informed decisions to apply normalization, smooth noisy variables, and engineer trend-based features from STL decomposition outputs to improve class transition prediction.

Boxplot visualizations were generated to examine the distribution of each sensor variable across different class labels (Normal, Transient Anomaly, and Steady Anomaly) for both real and simulated datasets. It was observed that some variables, such as P.MON.CKP and P.TPT showed clear shifts in their value ranges depending on the class. For example, in several wells, these variables had noticeably lower or higher median values during anomalous conditions compared to normal periods. This suggests that pressure and temperature readings are affected by transitions between normal and faulty states. On the other hand, variables such as P.PDG remained constant or had a single value in some wells, indicating either a lack of variation or issues with sensor reading. These findings provide useful guidance for feature selection in the next modeling stage, as not all variables contribute equally to the detection of anomalies.

The variable P.PDG was analyzed across all real datasets to check its distribution and relevance. In the datasets from WELL-00001 and WELL-00002, the P.PDG values were all found to be zero. This indicates that the sensor might not have been active or used during those data

collection periods. In the datasets from WELL-00006 (dated 20170731180930 and 20170731220432), P.PDG was recorded as a single constant value (6506.122) throughout the entire period. This lack of variation suggests that the sensor was either not responding to pressure changes or was providing fixed default readings. In the final dataset from WELL-00006 (dated 20180617200257), the P.PDG variable was either completely missing or contained only NA values. Based on these observations, it was decided that the P.PDG variable does not provide meaningful information and should be excluded from further analysis.

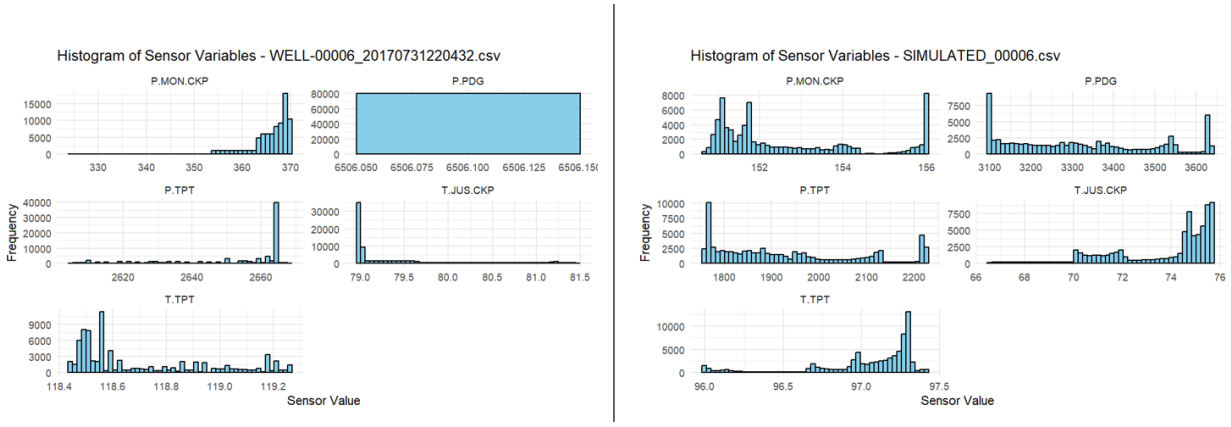

Figure 3-7. Comparison of Sensor Value Distributions: Real vs. Simulated Dataset (WELL-00006 vs. SIMULATED_00006)

Rolling Averages with Class Shading: A rolling average (also known as a moving average) is a smoothing technique that calculates the average of a variable over a defined window as it moves across the time series. This method reduces short-term noise while preserving long-term patterns in the data, making it especially effective for highlighting gradual transitions and structural changes in sensor signals (Hyndman & Athanasopoulos, 2018). A rolling average was applied to sensor data using a 60-second window to smooth short-term fluctuations and make general trends more visible. This technique helped highlight the transitions between normal, transient anomaly, and steady anomaly stages. In most wells, normal conditions showed stable patterns, while transient anomaly periods were marked by rapid changes in sensor values. After

these shifts, the values typically settled into new ranges during steady anomaly stages. The class labels matched well with the observed trends in the smoothed data. This confirmed that the rolling average approach was useful for visual exploration of anomaly progression in the time series.

STL Decomposition: STL decomposition was applied during the exploratory phase to dissect each sensor variable into three interpretable components: trend, seasonality, and remainder, allowing for a clearer understanding of how the signals evolve under different operational classes. The STL method was particularly suitable due to its robustness against noise, flexibility in handling non-stationary signals, and ability to adapt to time-varying seasonal patterns (RB, 1990). Visual analysis of the STL plots revealed that the trend components clearly aligned with class transitions. For example, in WELL-00001, both P.TPT (Figure 3-9) and P.MON.CKP exhibited a sharp downward trend during the steady anomaly stage, indicating persistent pressure reduction. Similarly, in SIMULATED_00001, the trend of P.TPT (Figure 3-8) showed a rapid increase during the transient anomaly, followed by stabilization, reflecting the system's shift to a steady anomalous condition. The seasonal components, especially in variables like T.JUS.CKP and P.MON.CKP displayed repeating patterns during the normal stage, but these patterns became distorted or dampened during anomaly stages, suggesting operational disruptions. The remainder components captured short-term spikes and noise, which intensified during transient anomalies. This was particularly evident in wells such as WELL-00006 and SIMULATED_00001, where P.TPT and T.TPT exhibited increased volatility. These fluctuations highlight the remainder's potential as an early indicator of instability. Additionally, consistent shifts in the trend and variance confirmed the non-stationary nature of most variables, which was supported by the results of the ADF stationarity tests. Therefore, STL decomposition not only aided in class-wise visual interpretation

but also informed the need for transformations (e.g., detrending or differencing) before applying deep learning models like LSTM or GRU.

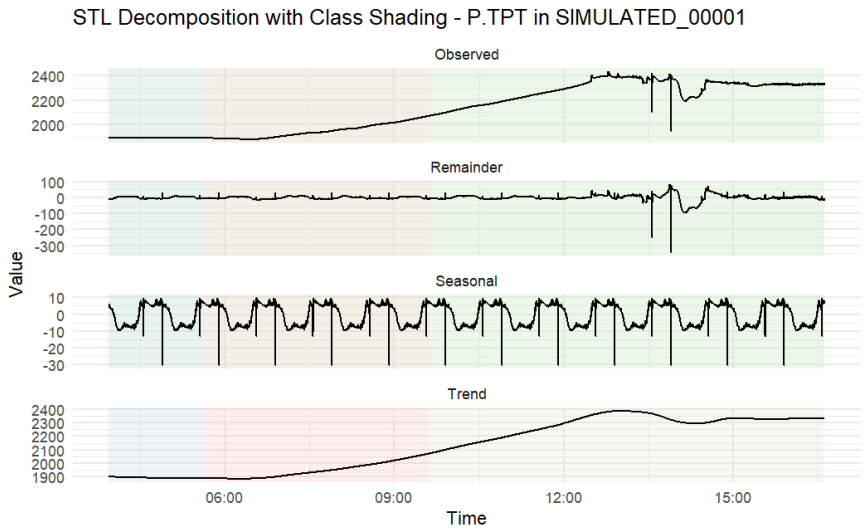

Figure 3-8. STL Decomposition of variable P.TPT (SIMULATED-00001)

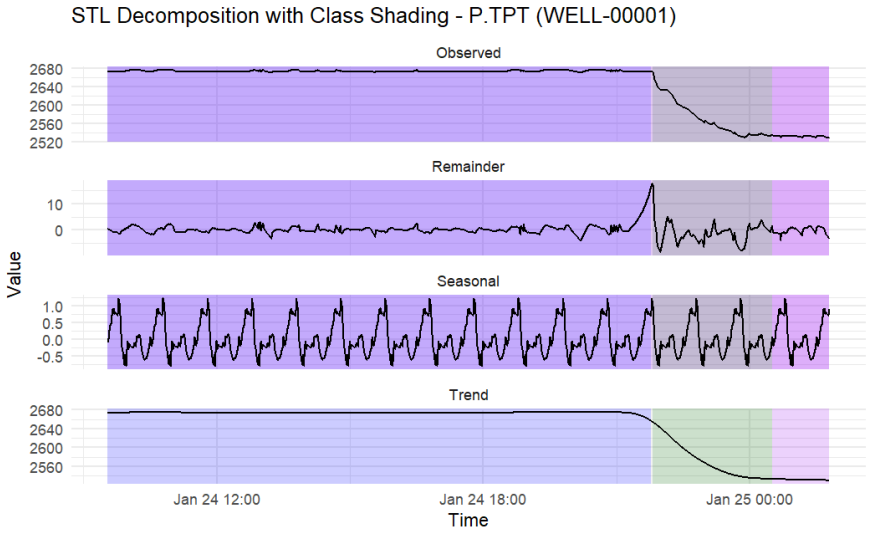

Figure 3-9. STL Decomposition of variable P.TPT (WELL-00001)

Stationarity Testing: Before applying time series modeling techniques, it was essential to assess the stationarity of the data, as many classical forecasting models—such as ARIMA—assume that the underlying time series has constant statistical properties over time. A stationary time series maintains a consistent mean, variance, and autocorrelation structure, which ensures

model stability and reliable forecasting. In contrast, non-stationary series can produce misleading or unstable results due to changing dynamics.

To evaluate this, the Augmented Dickey-Fuller (ADF) test was employed. The ADF test statistically assesses whether a unit root is present in a time series, with the null hypothesis indicating non-stationarity. A p-value below 0.05 typically suggests that the series is stationary, while a higher p-value implies that transformations such as differencing may be needed to stabilize the data before modeling (Dickey & Fuller, 1979).

In the analysis, the ADF test revealed that most real datasets were non-stationary. For example, in WELL-00001, all variables—P.TPT, T.TPT, and P.MON.CKP had p-values significantly above 0.05, confirming non-stationary behavior. A similar trend was observed across other real well datasets, with only a few exceptions (e.g., P.MON.CKP in WELL-00002 was stationary). Conversely, several simulated datasets showed stationary behavior in key variables. In SIMULATED_00002 and SIMULATED_00006, variables such as P.PDG, P.TPT, and T.TPT had p-values below 0.05, indicating stationarity.

These findings highlight the inherent non-stationarity of real-world sensor data, often caused by long-term trends, operational shifts, or anomalies. Recognizing this is crucial for determining the appropriate preprocessing steps—such as detrending or differencing—before applying time series models like ARIMA or even to inform the architecture and input format of neural models like LSTM and GRU.

During data preprocessing, missing values were handled conservatively to maintain the integrity of time series patterns. For visualization purposes, missing class labels were temporarily filled with a placeholder value (-1) to preserve continuity in class-shaded plots. Sensor variables

that were either entirely missing (e.g., P.PDG in WELL-00006) or constant throughout the dataset were excluded from the analysis, as they did not provide informative variation. No imputation was applied to partially missing sensor readings to avoid introducing artificial trends; instead, smoothing techniques like rolling averages were used to mitigate the impact of short-term gaps without distorting temporal behavior.

3.3.3 Application of Autoencoder-based Distortion

In this study, an autoencoder-based distortion technique was applied to enhance the realism of simulated offshore production data. Although Generative Adversarial Networks (GANs) (Brophy et al., 2023)were initially considered due to their ability to learn complex data distributions, early experiments produced unrealistic and unstable results. The synthetic outputs from GANs often failed to capture temporal structure and appeared overly noisy or collapsed to repetitive patterns. This was attributed to the limited amount of real training data available and computational resource constraints. As a result, GAN-based distortion was not pursued further in this thesis.

Instead, a denoising autoencoder trained exclusively on real well data was employed. This model learned the latent structure underlying typical sensor behavior, capturing temporal dependencies, anomaly transitions, and non-stationary patterns observed in real operational environments (Gensler et al., 2016). When applied to simulated data, the trained autoencoder reconstructed inputs by imposing learned characteristics—effectively distorting overly idealized and smooth synthetic signals into more realistic versions.

The transformation quality was evaluated using visual and statistical comparisons. For instance, in the P.TPT variable from the SIMULATED_00001 dataset, the distorted output showed clear deviation from its previously flat pattern—introducing localized fluctuations, shifts in

variance, and temporal noise consistent with real sensor behavior. These changes were structure-aware rather than random, indicating that the autoencoder effectively learned operational nuances such as transient disturbances and anomaly signatures.

Moreover, the method preserved temporal coherence, maintaining smooth transitions across consecutive time steps. Even though the plots used generic time indices, the underlying structure aligned with the real-time progression, ensuring the distorted signals remained usable for modeling. As a result, the generated data became more suitable for training supervised models like LSTM and GRU in downstream classification and forecasting tasks.

Notably, the autoencoder was not used as a classifier but as a feature-preserving transformer—demonstrating a hybrid approach where unsupervised learning enhanced supervised tasks. This aligns with recent literature emphasizing the use of autoencoders for semi-synthetic data enhancement and feature-aligned reconstruction in time-series applications (Gensler et al., 2016).

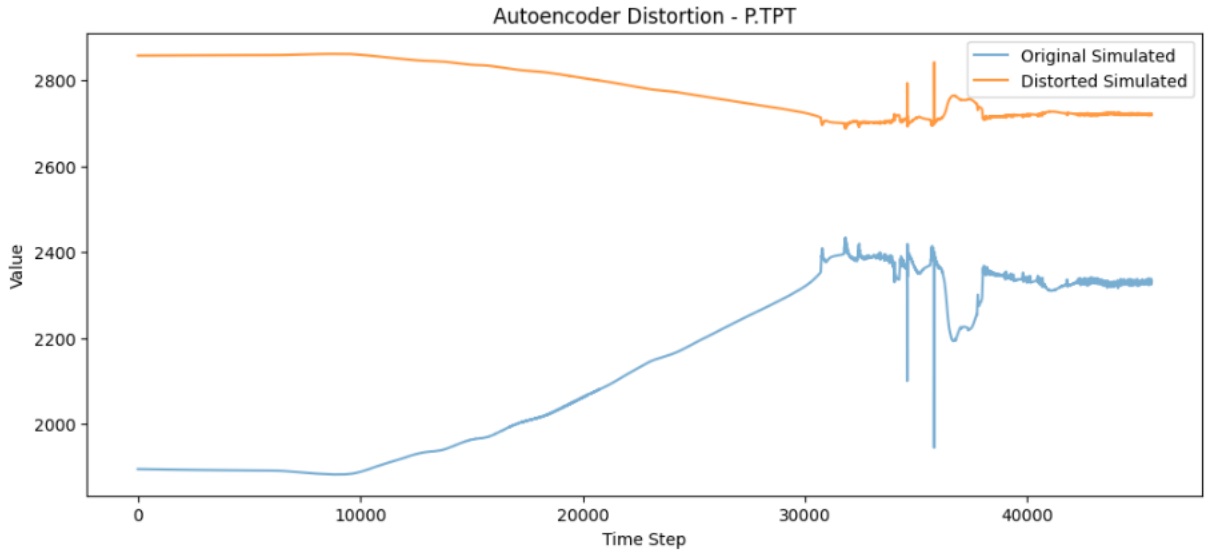

Figure 3-10. Visual Comparison of autoencoder distortion on variable P.TPT (SIMULATED_00001)

3.4 Modeling

Before initiating time-series modeling, several preprocessing and exploration steps were conducted to ensure data quality and modeling relevance. Real datasets from wells 1, 2, and 6—along with their corresponding simulated datasets—were selected based on their inclusion of anomaly label 1 (steady anomaly), which features a clear transition stage useful for prediction. Sensor variables were first checked for completeness; variables like P.PDG were excluded due to missing, constant, or uninformative values. Missing class labels were handled, and class distributions guided the focus on a specific anomaly type. To enhance realism, an autoencoder-based distortion was applied to the simulated datasets using patterns learned from real well behavior. STL decomposition and rolling averages were used to visualize trends, seasonal patterns, and anomaly transitions, revealing non-stationary behavior across most variables. These insights justified the use of sequential deep learning models—LSTM and GRU—which are well-suited for capturing long-term dependencies, class transitions, and nonlinear patterns in time-ordered sensor data.

3.4.1 LSTM

To perform multi-class anomaly classification in offshore oil production, a Long Short-Term Memory (LSTM) neural network was implemented. LSTM is a type of recurrent neural network (RNN) capable of learning long-term dependencies in sequential data, particularly effective for time-series tasks where past values influence future states (Hochreiter & Schmidhuber, 1997). Real well data (Wells 1, 2, and 6) and their corresponding distorted simulated datasets (Simulated 1, 2, and 6) were used for model training and evaluation. Simulated datasets were enhanced using an autoencoder-based distortion technique, which applies a

denoising autoencoder trained on real data to reconstruct synthetic inputs into more realistic forms by embedding learned temporal dynamics (Vincent et al., 2008)

Importantly, the model was trained exclusively on distorted simulated datasets and evaluated on real well datasets. This approach was designed to assess the model's ability to generalize from synthetic data to actual operational conditions, which is critical for real-world deployment where labeled real data may be limited or unavailable for training.

To prepare the data for modeling, MinMaxScaler was applied to normalize each feature between 0 and 1, a common preprocessing step that improves convergence and stability in neural networks. The original anomaly class labels—Normal (0), Steady Anomaly (1), and Transient Anomaly (101)—were recoded into sequential integers (0, 1, 2) and one-hot encoded for use with categorical cross-entropy loss in multi-class classification.

To address severe class imbalance, which can bias model training toward majority classes, class weights were computed based on label frequencies and incorporated into the loss function (He & Garcia, 2009). The LSTM model architecture consisted of 64 LSTM units, a dropout layer (rate = 0.3) to reduce overfitting by randomly ignoring connections during training, a dense hidden layer with 32 neurons using ReLU activation, and a final softmax output layer to generate probabilities for each of the three classes. For model optimization, Keras Tuner was employed with RandomSearch, a technique that explores a defined hyperparameter space to identify the most performant configuration (Li et al., 2020). The search included variations in the number of LSTM units, dense layer sizes, dropout rates, and learning rates across 10 trials.

3.4.2 GRU

In this phase of the study, a Gated Recurrent Unit (GRU) neural network was developed to classify multivariate time series data into three anomaly categories: Normal operation (0), Steady Anomaly (1), and Transient Anomaly (101). GRUs are a type of recurrent neural network (RNN) designed to efficiently capture long-term dependencies in sequential data while being less computationally intensive than traditional LSTMs (Chung et al., 2014). The model was trained using a combination of all autoencoder-distorted simulated datasets and tested on real well datasets.

This design—training on distorted simulated data and testing on real data—was crucial for ensuring model generalization. It allowed the GRU to learn from a broader operational landscape while being evaluated on realistic conditions that reflect true production behavior. This separation also avoided data leakage and maintained the integrity of temporal dependencies.

MinMaxScaler was used to normalize sensor values between 0 and 1. The normalized features were reshaped into a 3D format compatible with GRU input: (samples, time steps, features). Class imbalance was addressed by calculating class weights from Scikit-learn and incorporating them into model training. The GRU model architecture consisted of a single 64-unit GRU layer, followed by a dropout of 0.3 for regularization, a dense hidden layer with 32 ReLU units, and a softmax output layer for multi-class classification. The model was compiled with the Adam optimizer and trained for 10 epochs with a batch size of 64, using 20% of the training data for validation.

3.4.3 Regression-based LSTM Classification

In the final stage of this study, a regression-based time series modeling approach was adopted to anticipate upcoming system behaviors and detect anomalies before they fully

developed. Unlike standard classification, this method frames the problem as a sequence-to-one regression task, where historical sequences of multivariate sensor measurements are used to predict the next class-like value. A Long Short-Term Memory (LSTM) neural network was employed due to its ability to capture long-term temporal dependencies and handle non-linear dynamics in multivariate time series data (Hochreiter & Schmidhuber, 1997). The model was trained on all three distorted simulated datasets and tested on the real datasets to preserve temporal coherence and evaluate performance in real-world conditions.

To make the model learn temporal class evolution, class labels were temporally redefined: Class 0 (Normal) as time step t−1, Class 101 (Transient Anomaly) as t, and Class 1 (Steady Anomaly) as t+1, effectively embedding sequence awareness into the training structure. The dataset was preprocessed into sequences using a window size of 3, meaning each input sample consisted of 3 consecutive time steps of data used to predict the next class. This sliding window technique allows the model to learn how operational states transition over time (Brownlee, 2017).

The final LSTM architecture included an LSTM layer with 64 memory units, followed by a Dropout layer (0.3) to prevent overfitting by randomly deactivating 30% of the neurons during training. A Dense layer with 32 ReLU-activated neurons served as an intermediate mapping layer before the final output layer with one neuron and linear activation, suitable for regression tasks. The model was trained for 10 epochs, using a batch size of 64, which defines how many samples are processed at once before updating model weights (Goodfellow, Bengio, Courville, et al., 2016). The Adam optimizer was used for efficient and adaptive learning, and Mean Squared Error (MSE) was chosen as the loss function due to its effectiveness in continuous regression scenarios.

The model produced continuous numerical outputs, which were rounded and clipped between 0 and 2 to map them back to class labels. For example, an output of 1.87 was clipped to

2, indicating a high likelihood of a transient anomaly. This soft regression mechanism enabled more flexible learning compared to rigid classification, especially for predicting gradual shifts in well conditions.

Chapter 4 : RESULTS and DISCUSSION

This chapter presents the outcomes of the machine learning models applied to classify and detect undesirable events in offshore oil well production based on the 3W Petrobras dataset. Comparisons are made between traditional classifiers—such as KNN, Decision Tree, Random Forest, and Support Vector Machines—and sequence-based models like LSTM and GRU, which were trained on temporally structured data.

4.1 Supervised Classification

4.1.1 K-Nearest Neighbor (KNN)

The final model with 10 neighbors was evaluated on the test dataset using metrics such as accuracy (90%), precision, recall, and F1-score. A confusion matrix was generated to analyze class-wise prediction performance. Precision (Figure 4-1(left)) and recall (Figure 4-1(right)) values for varying K values were plotted for each class. It can be observed that most classes maintain high precision and recall across different K values, especially Class 1 and Class 3, indicating consistent detection performance. However, Class 6 shows a significant decline in both precision and recall beyond K=20, suggesting that the model fails to correctly identify this class when K increases. This may be due to class imbalance or feature overlap with other classes. These plots highlight the importance of selecting an optimal K value—too high a value may smooth out boundaries and hurt minority class performance, while too low a K may lead to overfitting.

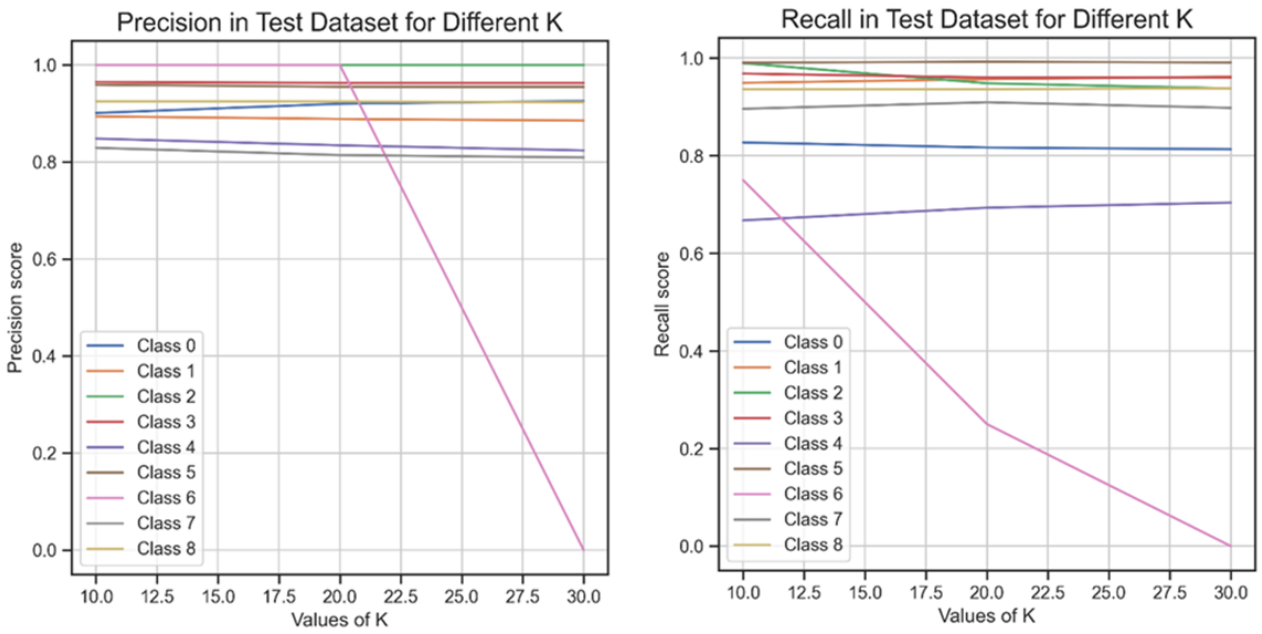

Figure 4-1. Precision (left), Recall (right) for KNN Classification

The confusion matrix in Figure 4-2 provides a detailed view of how well the KNN classifier performed across all classes. The diagonal elements represent correct predictions, while off-diagonal elements indicate misclassifications. The model showed strong predictive accuracy for most classes—particularly for Class 5 (Rapid Productivity Loss) and Class 4 (Flow Instability), with high counts of correct predictions (3032 and 257, respectively).

However, Class 6 (Quick restriction in Production Choke) suffered from misclassification, which is likely due to class imbalance—this class had only 23 samples, making it difficult for the KNN algorithm to learn reliable decision boundaries. Similar issues were observed for Classes 7 and 8, although their performance was relatively better. The consistency of F1 scores during 10-fold cross-validation indicates that the model generalizes well and does not overfit the training

data. These results highlight the importance of both class distribution and feature distinguishability in supervised learning (Japkowicz & Stephen, 2002).

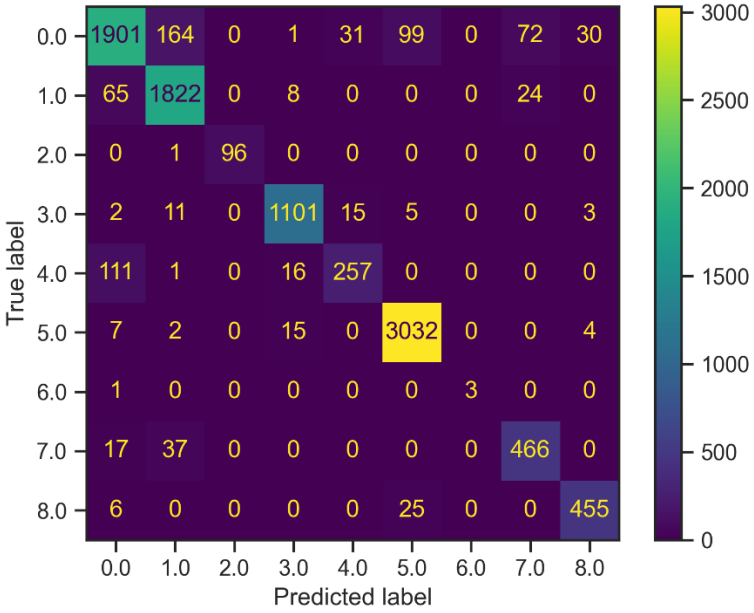

Figure 4-2. Confusion Matrix for KNN Classification

4.1.2 Decision Tree (DT)

The Decision Tree (DT) model was evaluated on the test dataset using metrics such as accuracy (95%), precision, recall, and F1-score. As shown in the confusion matrix (Figure 4-3), the model performed strongly across most classes, especially for majority classes like Class 0, Class 1, and Class 5, where predictions closely matched the true labels. Misclassifications were minimal and mostly observed in underrepresented classes such as Class 6 (Quick restriction in Production Choke) and Class 8 (Hydrate formation), which is expected due to their smaller sample sizes. Despite this, the overall structure of the matrix shows that most of the diagonal elements (correct predictions) have high counts compared to off-diagonal elements (errors), indicating high predictive accuracy.

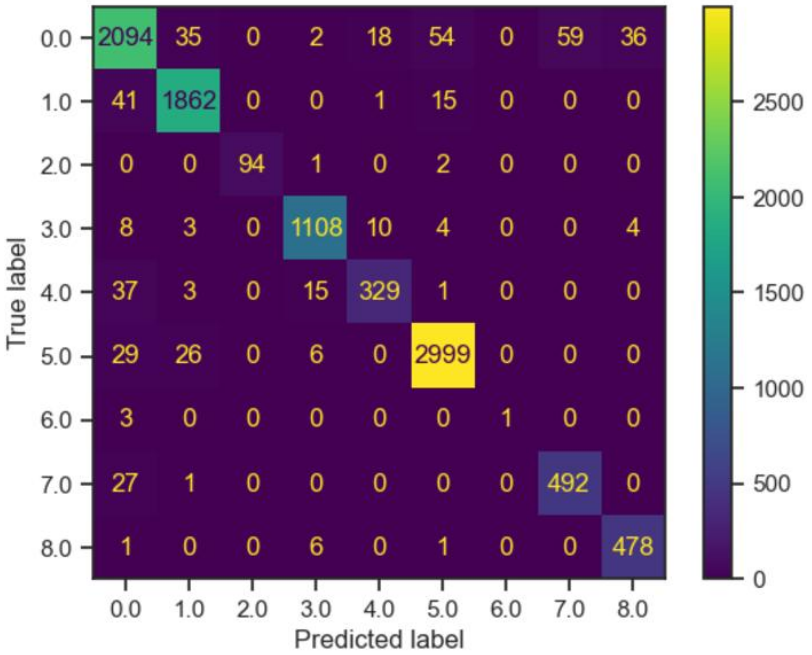

Figure 4-3. Confusion Matrix for DT Classification

Additionally, the F1-score plot from 10-fold cross-validation (Figure 4-4) demonstrates that the DT classifier yielded consistently high performance across all folds, with F1 scores very close to 1. This confirms the model's stability and generalizability across different data partitions. One of the advantages of using DTs is that they do not require feature scaling since they are invariant to monotonic transformations and unaffected by the magnitude of input variables.

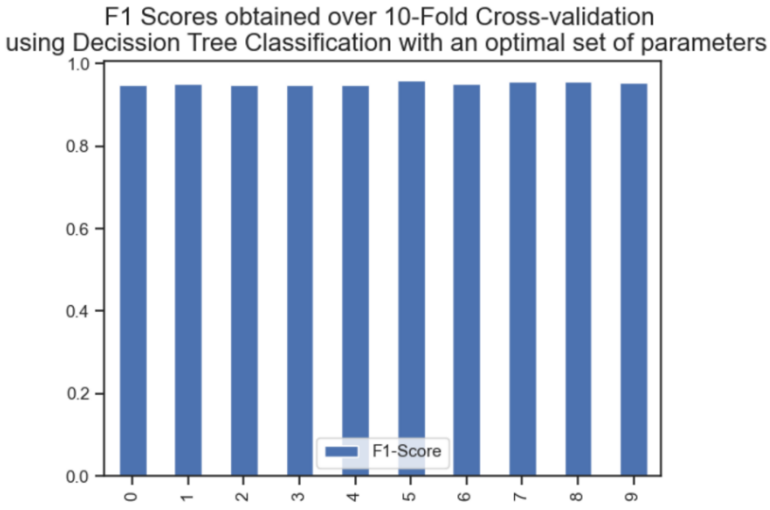

Figure 4-4. F1 score obtained over 10-fold cross-validation for DT Classification

Therefore, unlike models like KNN or SVM, preprocessing steps like normalization were not necessary in this case. DT's robustness, interpretability, and low computational complexity make it a suitable baseline classifier for the multi-class prediction task in offshore event detection.

4.1.3 Random Forest (RF)

The model achieved high accuracy with 94% and consistent F1 scores across cross-validation folds. In the classification report of RF, the model maintained strong precision and recall values across almost all classes, with particularly high F1-scores for Classes 1 through 5. Although Class 6 had perfect precision, it suffered from a very low recall (0.25), which is likely due to a highly imbalanced dataset with only 4 instances of this class. The confusion matrix (Figure 4-5) supports this observation, as most misclassifications occurred in the minority classes. Despite this, the overall weighted F1-score remained 0.90, demonstrating the model's robustness and ability to generalize across most event categories. Random Forest's ensemble nature allowed it to manage variance and improve predictive stability, making it a strong candidate for real-time offshore anomaly detection tasks.

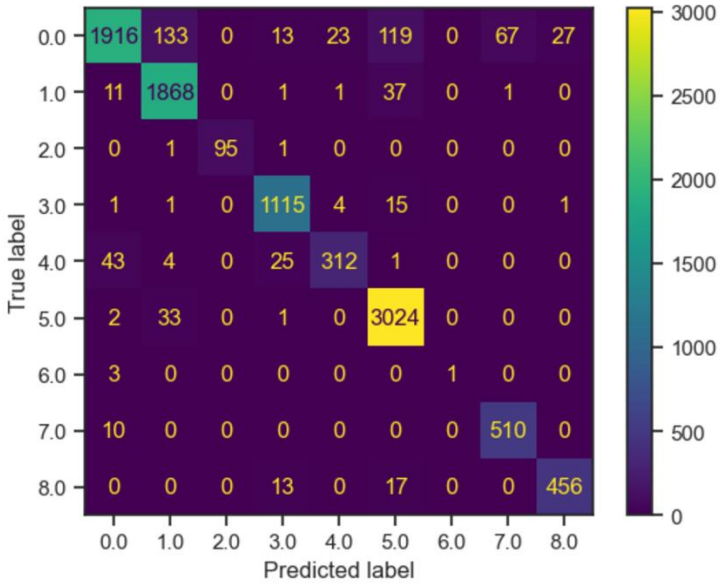

Figure 4-5. Confusion Matrix for KNN Classification

4.1.4 Neural Networks (MLP)

Using cross-validation, various configurations of hidden layer sizes were tested, and the best-performing configuration was determined to be two layers with 25 neurons each. This configuration achieved the highest mean test score across validation splits, indicating strong generalization capability.

The final neural network model (Multilayer Perceptron Classifier, MLP) achieved an overall accuracy of 90% on the test dataset. The model demonstrated high F1-scores across most classes (e.g., Class 2 and Class 3 above 0.95), confirming its effectiveness in handling multi-class classification. However, Class 6, which had significantly fewer samples, showed poor recall (0.25), highlighting the model's sensitivity to class imbalance.

The confusion matrix during analysis further confirms this behavior, where most predictions closely align with true labels except for minority classes, which were occasionally misclassified into majority categories. Despite this, the model maintained balanced performance across other classes, as reflected by the weighted average F1-score of 0.90.

4.1.5 Support Vector Machine (SVM)

The SVM model achieved an overall accuracy of 92%, with a weighted average F1-score and precision both at 0.92, confirming the model's robustness across all classes. As shown in the confusion matrix (Figure 4-6 (right)), the classifier predicted most class labels correctly, with minor misclassifications in classes such as Class 0 and Class 4, which had slightly more overlap with neighboring categories. Notably, the classifier completely failed to predict Class 6, likely due to the class's underrepresentation in the dataset. Despite this, the model demonstrated excellent recall in classes with strong support, like Class 1, Class 3, and Class 5. The bar plot in Figure 4-6 shows consistently high F1 scores across 10 cross-validation folds, indicating model stability.

These results highlight the effectiveness of the RBF kernel in separating non-linearly distributed class boundaries while maintaining generalization. SVM's strong performance across well-represented classes validates its suitability for complex multi-class classification problems in time-sensitive industrial applications (Hsu et al., 2003)

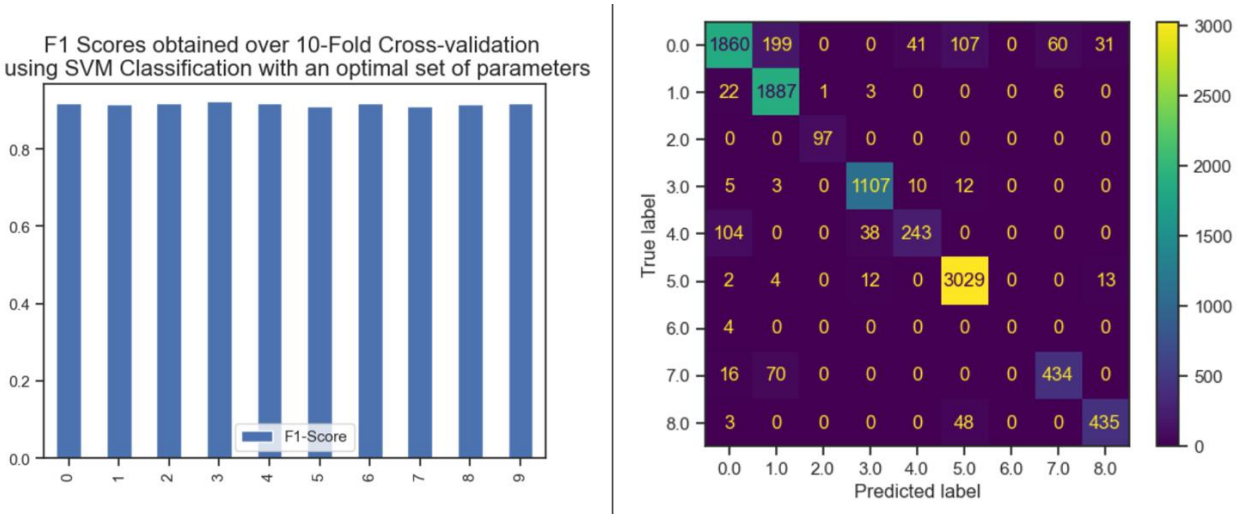

Figure 4-6. F1 scores obtained over 10-fold Cross-validation using SVM Classification (left), Confusion Matrix for SVM Classification (right)

4.1.6 Summary of Classification Results

Table 4-1, given below, provides a summary of the four supervised classification machine learning models. In general, all models achieved high accuracy.

The Decision Tree (DT) model is the best-performing model among the four supervised classification techniques. It achieved the highest accuracy (0.95), the highest macro average F1 score (0.87), and the highest weighted average F1 score (0.95). These results demonstrate that the Decision Tree outperformed the other models in terms of overall classification performance. It is not only exhibited the highest accuracy but also required the least computational time and power compared to the other models.

In contrast, the Random Forest and Neural Network models were the most computationally expensive and time-consuming to run. This is due to the nature of these algorithms and the additional time required for hyperparameter optimization, which is integral to their performance improvement.

Table 4-1. Model Performance Comparison of Supervised Classification Models

Model	Accuracy	Macro Avg (F1)	Weighted Avg (F1)
KNN	0.91	0.80	0.91
Decision Tree (DT)	0.95	0.87	0.95
Random Forest (RF)	0.94	0.86	0.94
Neural Network (MPL)	0.90	0.81	0.90
Support Vector Machine (SVM)	0.92	0.81	0.92

Additionally, for this modeling approach, data imbalance was not specifically addressed since the macro average F1-score for each model exceeds 80%, which is considered highly promising. The best-performing model, the Decision Tree, achieved a macro average F1-score of 0.87. The macro average treats all classes equally by assigning equal weight to each, regardless of their frequency. Therefore, achieving such high scores despite the presence of class imbalance indicates strong overall model performance. The results highlight the trade-off between computational cost and model complexity.

4.2 Time-Series Modeling Approach

This section presents the results of time-series-based classification models developed to detect operational anomalies in offshore oil production data. Detailed performance evaluations, including confusion matrices and classification metrics, are provided to assess how effectively each model distinguishes between Normal, Transient Anomaly, and Steady Anomaly classes.

4.2.1 LSTM-based Classification

The final LSTM model achieved a best validation accuracy of 83.16% after hyperparameter tuning with Keras Tuner. On the test set, the model demonstrated strong capability in detecting Normal operation, with a precision of 0.74 and a perfect recall of 1.00, correctly classifying all 152,246 Normal instances. Detection of Steady Anomaly was moderate, with a recall of 0.50, while the model faced greater difficulty in correctly identifying Transient Anomalies, achieving a recall of 0.35.

The confusion matrix (Figure 4-7) reveals a clear challenge in distinguishing Transient Anomalies from other classes. A significant number of Transient Anomaly instances (53,423) were misclassified as Normal, and 4,828 Steady Anomaly cases were also confused with Transient Anomaly. This misclassification reflects the temporal overlap and subtle transitions during anomaly onset stages, where signal patterns may still resemble normal conditions or intermediate behaviors. However, this challenge is not entirely negative—the fact that Transient Anomaly was often confused with Steady Anomaly suggests the model is sensitive to early-stage deviations. Since transient stages often precede steady failures, detecting them—even if misclassified—may still serve as a useful warning signal for operators.

Despite the complexity of multivariate time series and class imbalance, the model achieved an overall accuracy of 75.5% and a weighted F1-score of 0.72, indicating strong generalization. Class weighting contributed to improved detection across minority classes, and the use of autoencoder-based distortion helped the model better adapt to realistic signal patterns in the test data. Overall, the LSTM architecture successfully captured sequential dependencies and temporal transitions, proving suitable for early anomaly detection in offshore well sensor data.

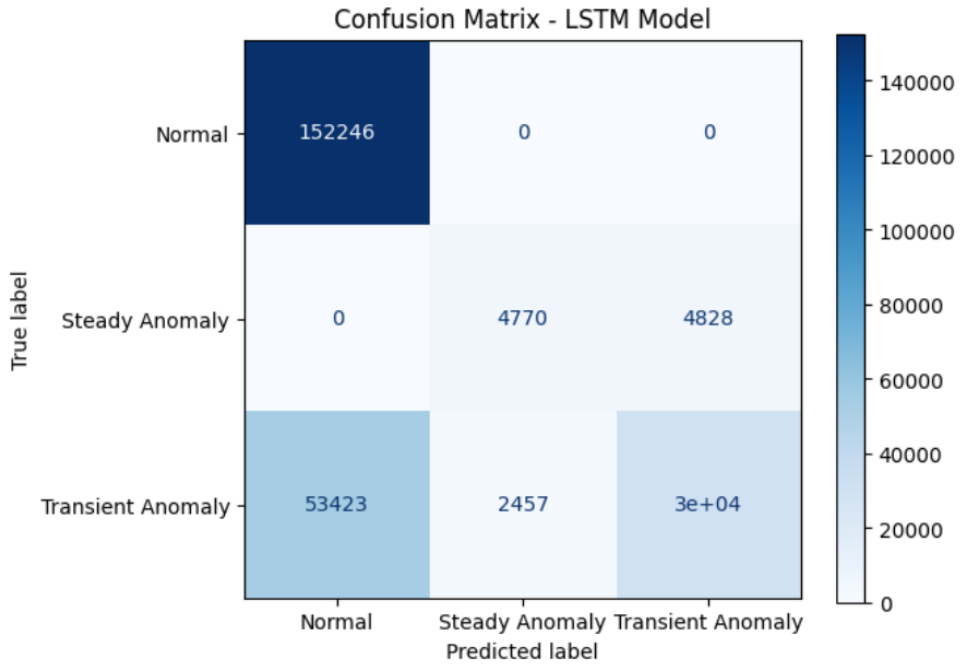

Figure 4-7. Confusion Matrix for LSTM

4.2.2 GRU-based Classification

The final GRU model achieved strong classification performance on the real test data, with an overall accuracy of 89.55% and a macro-average F1-score of 89.77%, indicating balanced predictive ability across all anomaly classes. Class-wise evaluation revealed that Class 0 (Normal) achieved the highest precision at 97.26%, with a recall of 86.59%, showing the model's effectiveness in correctly identifying stable operating conditions. Class 1 (Steady Anomaly) recorded a recall of 95.90%, confirming the model’s strength in capturing persistent deviations in sensor behavior. For Class 101 (Transient Anomaly), the model achieved 84.27% precision and 89.32% recall, demonstrating its capability to detect complex transitional states that often overlap with normal operation. These results suggest that the GRU model handled class imbalance and non-linear signal transitions effectively.

The improved performance compared to the LSTM model can be attributed to the GRU’s simplified gating mechanism, which reduces the risk of vanishing gradients while still preserving

important sequence information. Additionally, the model was trained using a combination of all distorted simulated datasets and tested on merged real well datasets. Although datasets were merged, the GRU model was trained using single time-step inputs, ensuring no artificial sequence continuity was introduced across different wells. This setup preserved sequence integrity and enabled the model to generalize well across diverse operational patterns. The realistic variability introduced through autoencoder-based distortion further enhanced the model's ability to learn nuanced anomaly signatures. Overall, the GRU-based classifier proved highly effective in multi-class anomaly detection and confirmed the benefits of combining distorted synthetic data for training robust time series models.

4.2.3 Regression-based LSTM Classification

The regression-based LSTM model demonstrated strong performance in predicting time-evolving system behavior, particularly in capturing transitional and anomaly states. Unlike traditional classification, this approach framed the task as a sequence-to-one regression problem, using historical sensor measurements to forecast a continuous class-like value.

The model was trained on distorted simulated datasets and tested on real well datasets, preserving temporal sequence integrity. It achieved an overall accuracy of 91.38%, outperforming the earlier classification-based LSTM model, which had an accuracy of 75.5%. Class-wise results were also notable: Class 0 (Normal) had a recall of 0.9997, indicating strong detection of stable states; Class 1 (Steady Anomaly) achieved a precision of 0.9283 and recall of 0.8690, confirming the model's effectiveness in identifying persistent deviations; and Class 2 (Transient Anomaly), typically the most challenging, was detected with 0.9976 precision and 0.9361 recall, showing robust recognition of subtle transitional behaviors.

These results validate the advantages of regression modeling in capturing the dynamic and evolving nature of operational anomalies. By allowing soft transitions rather than rigid class boundaries, the model improved both accuracy and interpretability—critical for timely, real-world anomaly detection in offshore production systems.

4.3 Summary

Among the time-series approaches tested in this study, the LSTM-based regression model demonstrated the best overall performance. It achieved the highest accuracy (91.4%), weighted F1-score (91.8%), and macro-average F1-score (88.6%). This model was especially effective in forecasting anomaly transitions using a continuous output strategy that was later rounded to discrete class labels. While the GRU model also performed strongly with a macro F1-score of 89.8% and better recall for steady anomalies (95.9%), the regression-based LSTM offered improved interpretability and smoother transition detection. The LSTM classification model, in contrast, struggled with transient and steady anomaly detection, particularly due to class imbalance and temporal complexity. Overall, the LSTM regression approach is recommended for early warning and anomaly forecasting in offshore well operations due to its superior performance and predictive capability (Table 4-2).

Table 4-2. Summary of Time-Series Modeling Results

Model	Accuracy	Macro F1-score	Weighted F1-score	Best Class Recall
LSTM (Classification)	75.5%	63.8%	71.7%	100.0% (Normal)
GRU	89.6%	89.8%	89.6%	95.9% (Steady Anomaly)
LSTM (Regression)	91.4%	88.6%	91.8%	99.9% (Normal)

Chapter 5 : CONCLUSION

This thesis investigated the application of supervised machine learning and time-series models for predicting undesirable production events in offshore oil wells using the 3W Petrobras dataset. The study explored the use of various supervised classifiers—KNN, Decision Tree, Random Forest, Artificial Neural Networks, and SVM—as well as sequential models like LSTM and GRU. A novel regression-based forecasting approach was also developed to model the transition between normal, transient, and anomalous operational phases.

A key contribution of this work was the integration of autoencoder-based distortion techniques for enhancing simulated time-series data. By learning the dynamic structure of real well behavior through autoencoders and transferring these patterns to simulated data, the distorted datasets more accurately reflected real-world conditions. This improved the generalizability and realism of models trained on synthetic examples, especially for rare events with limited real data.

The thesis also proposed a hybrid time-series regression-based LSTM classification framework in which LSTM models were trained to predict upcoming operational states based on past temporal patterns. In this approach, class transitions were modeled as continuous numerical outputs (regression), which were then discretized into class labels. This hybrid formulation enabled forward-looking classification that captures subtle transitions between operational phases—Normal (Class 0), Transient (Class 101), and Steady Anomaly (Class 1)—providing earlier warnings and more contextual understanding of event evolution.

The research has demonstrated that combining supervised classification with advanced time-series modeling significantly enhances the accuracy and interpretability of anomaly prediction. Among the classifiers, Decision Trees and Random Forests showed strong

performance, benefiting from their ability to model non-linear relationships and feature interactions. Meanwhile, recurrent neural network architectures effectively captured temporal dependencies, enabling earlier detection of evolving anomalies.

Preprocessing strategies such as feature scaling, outlier removal, and the injection of Gaussian noise were instrumental in improving model robustness. The use of autoencoder-informed distortion further enhanced data diversity, allowing models to generalize better despite the scarcity of real anomaly instances.

In conclusion, the proposed modeling pipeline provides a practical and scalable solution for real-time monitoring of offshore production systems. By integrating temporal modeling, supervised learning, synthetic data enhancement, and regression-based classification, the framework offers a forward-looking and data-driven approach to managing operational risks, improving safety, and minimizing downtime in offshore oil production environments.

Chapter 6 : RECOMMENDATIONS AND FUTURE WORK

6.1 Recommendations

Use a Combined Monitoring System: It is recommended to implement a hybrid monitoring system that uses both traditional rule-based alarms (like pressure thresholds) and machine learning predictions. This combination allows engineers to cross-check alerts, improving reliability while benefiting from the early detection of potential issues.

Keep the Models Updated with New Data: As new operational data becomes available, the machine learning models should be retrained regularly. This helps the system stay accurate and adapt to changes in well conditions, equipment behavior, or operational procedures.

Make the Model Easy to Understand for Engineers: To gain the trust of field operators, it's important to explain why the model predicted an anomaly. Tools like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) can be used to show which sensor values most influenced the prediction (Ribeiro et al., 2016).

Connect the Models to Real-Time Systems: For full value, the models should be connected to real-time monitoring platforms such as SCADA (Supervisory Control and Data Acquisition) systems or digital twins—virtual representations of wells. This allows engineers to take immediate action if the system predicts a failure (Tao et al., 2018).

Work Together to Improve Anomaly Labels: Accurate labeling of anomalies is critical for training reliable models. Data scientists should collaborate closely with oil and gas engineers to refine how events—especially transitions and early warnings—are labeled. This will help reduce confusion in model training and improve accuracy.

6.2 Future Work

Expansion to Unsupervised and Semi-Supervised Learning: Due to the scarcity of labeled real events, future studies should explore unsupervised anomaly detection and semi-supervised learning to leverage unlabeled data more effectively.

GAN-Based Simulation Enhancements: Future experiments can further utilize Generative Adversarial Networks (GANs) to create more sophisticated simulated anomaly patterns, improving the realism of training data.

Multi-Well Modeling: A deeper investigation into generalizing models across different wells could be pursued to ensure scalability and transferability of the approach to new fields or operators.

Sensor Fusion and Additional Features: Integrating other sensor modalities (e.g., vibration, acoustic, or chemical sensors) and environmental data (e.g., sea temperature) could improve detection sensitivity.

Online Learning and Drift Detection: Incorporating online learning techniques and concept drift detection mechanisms would allow the model to adapt in real time to operational changes or new anomaly types.

Validation with Industrial Partners: Collaborative pilot projects with oil and gas companies could validate the models in live offshore environments, providing insights into deployment challenges and further optimization.

References

Aldabagh, H., Zheng, X., & Mukkamala, R. (2023). A hybrid deep learning approach for crude oil price prediction. *Journal of Risk and Financial Management*, *16*(12), 503.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175-185.

Andreolli, I. (2016). Introdução à elevação e escoamento monofásico e multifásico de petróleo. *Rio de Janeiro: Interciência*.

Arslan, M., Guzel, M., Demirci, M., & Ozdemir, S. (2019). SMOTE and gaussian noise based sensor data augmentation. 2019 4th international conference on computer science and engineering (UBMK),

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Brophy, E., Wang, Z., She, Q., & Ward, T. (2023). Generative adversarial networks in time series: A systematic literature review. *ACM Computing Surveys*, *55*(10), 1-31.

Brownlee, J. (2017). *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery.

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. Proceedings of the 23rd international conference on Machine learning,

Carvalho, B. G. (2021). Evaluating machine learning techniques for detection of flow instability events in offshore oil wells. *Master's degree dissertation*.

Cengel, Y. A., & Boles, M. A. (2002). Thermodynamics: an engineering approach. *Sea*, *1000*(8862), 287-293.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition. *J. off. Stat*, *6*(1), 3-73.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273-297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

CSB. (2016). Investigation Report Executive Summary. Drilling Rig Explosion and Fire at the Macondo Well. In: US Chemical Safety and Hazard Investigation Board Washington, DC, USA.

CSB, U. (2014). Explosion and fire at the Macondo well. *US Chemical Safety and Hazard Investigation Board, Investigation Report*, 2.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, *74*(366a), 427-431.

Feng, R., Zhao, D., & Zha, Z.-J. (2021). Understanding noise injection in gans. international conference on machine learning,

Fernandes Jr, W., Komati, K. S., & Assis de Souza Gazolli, K. (2024). Anomaly detection in oil-producing wells: a comparative study of one-class classifiers in a multivariate time series dataset. *Journal of Petroleum Exploration and Production Technology*, *14*(1), 343-363.

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, *32*(14-15), 2627-2636.

Gensler, A., Henze, J., Sick, B., & Raabe, N. (2016). Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. 2016 IEEE international conference on systems, man, and cybernetics (SMC),

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. Proceedings of the fourteenth international conference on artificial intelligence and statistics,

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Regularization for deep learning. *Deep learning*, 216-261.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

Hajirahimi, Z., & Khashei, M. (2019). Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence*, *86*, 83-106.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263-1284.

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

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A practical guide to support vector classification. In: Taipei, Taiwan.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(1), 4-37.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, *6*(5), 429-449.

Jiang, Q., Yan, X., & Huang, B. (2019). Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes. *Industrial & Engineering Chemistry Research*, *58*(29), 12899-12912.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Ijcai,

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). Springer.

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B., & Talwalkar, A. (2020). A system for massively parallel hyperparameter tuning. *Proceedings of machine learning and systems*, *2*, 230-246.

Malhotra, G., Leslie, D. S., Ludwig, C. J., & Bogacz, R. (2017). Overcoming indecision by changing the decision boundary. *Journal of Experimental Psychology: General*, *146*(6), 776.

Marins, M. A., Barros, B. D., Santos, I. H., Barrionuevo, D. C., Vargas, R. E., Prego, T. d. M., de Lima, A. A., de Campos, M. L., da Silva, E. A., & Netto, S. L. (2021). Fault detection and classification in oil wells and production/service lines using random forest. *Journal of Petroleum Science and Engineering*, *197*, 107879.

Mohaghegh, S. (2000). Virtual-intelligence applications in petroleum engineering: Part 1—Artificial neural networks. *Journal of Petroleum Technology*, *52*(09), 64-73.

Nikitin, N. O., Revin, I., Hvatov, A., Vychuzhanin, P., & Kalyuzhnaya, A. V. (2022). Hybrid and automated machine learning approaches for oil fields development: The case study of Volve field, North Sea. *Computers & geosciences*, *161*, 105061.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*, 81-106.

RB, C. (1990). STL: A seasonal-trend decomposition procedure based on loess. *J Off Stat*, *6*, 3-73.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining,

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3), e0118432.

Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis,

Schlumberger, O. (2014). Dynamic multiphase flow simulator. In: Version.

Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of manufacturing systems*, *48*, 157-169.

Theyab, M. A. (2018). Fluid flow assurance issues: literature review. *SciFed Journal of Petroleum*, *2*(1), 1-11.

Turan, E. M., & Jäschke, J. (2021). Classification of undesirable events in oil well operation. 2021 23rd international conference on process control (PC),

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., & 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.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning,

Yacouby, R., & Axman, D. (2020). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. Proceedings of the first workshop on evaluation and comparison of NLP systems,

Zarinabadi, S., & Samimi, A. (2012). Problems of hydrate formation in oil and gas pipes deals. *Journal of American Science*, *8*(8), 1007-1010.