Master Thesis

Comparative Evaluation of Vibration Monitoring Techniques for Water Injection Systems of Floating Production Storage and Offloading Unit (FPSO)

Yikun Tang

Delft University of Technology

Master Thesis

Comparative Evaluation of Vibration Monitoring Techniques for Water Injection Systems of Floating Production Storage and Offloading Unit (FPSO)

by

Yikun Tang

Student Name	Yikun Tang
Student Number	5792266

University Supervisor:	Dr. Eliz-Mari Lourens, Dr. Hongrui Wang
Company Supervisors:	Vitor Castilho, Arnout Roos
Project Duration:	March, 2024 - Aug, 2024
Faculty:	Civil Engineering and Geosciences, TU Delft
	Operational Intelligence and Performance Optimization Center, SBM Schiedam B.V.

TU Delft

Preface

As I poised to pen the final words of this thesis, I first wish to extend my heartfelt gratitude to my four mentors at TU Delft and SBM Offshore. Without your guidance, this project would not have reached its fruition. The past year has been a whimsical journey for me, one that I never envisioned before arriving in the Netherlands, where I was to complete my master education across two esteemed institutions.

TU Delft has been a sanctuary of academia, where I imbibed foundational knowledge and theories, while SBM Offshore has been a crucible of practicality, where I applied my learnings and discerned the inner workings of an engineering enterprise. Besides, it is in these two places, through the teachings of my mentors and the camaraderie of classmates and colleagues, that I have gleaned the most profound lessons of life—how to get along with others and how to find harmony within myself.

Furthermore, I am beholden to my friends and my five roommates. Without their solace and care during my moments of solitude and helplessness, I shudder to think how I would have navigated this endeavor. Even as we may disperse across the globe, the memories of our gatherings, our marathon phone calls, and the myriad of experiences we have shared shall forever be etched in my mind.

Lastly, words fall short in expressing my gratitude to my parents. Their unwavering support and sacrifice have shaped me into the person I am today. *I love you, that's all, and more than that.*

Yikun Tang
Delft, September 2024

Abbreviation

AIC	Akaike Information Criterion
CDI	FPSO Cidade de Ilhabela
CDM	FPSO Cidade de Maricá
CDS	FPSO Cidade de Saquarema
ESD	Emergency Shut Down
FI	Flow Indicator
FIC	Flow Indicating Controller
FPSO	Floating, Production, Storage and Offloading Unit
FSI	Flow Safety Indicator
GTG	Gas Turbine Generator
HP	High Pressure
IEEE	Institute of Electrical and Electronics Engineers
KE	Keyphasor Probe
KMO	Kaiser–Meyer–Olkin test
LI	Level Indicator
LSI	Level Safety Indicator
LSTM	Long Short-Term Memory
MGC	Main Gas Compressor
MSE	Mean Square Error
OIPOC	Operational Intelligence and Performance Optimization Center
PCA	Principal Component Analysis
PDI	Pressure Differential Indicator
PI	Pressure Indicator
PM	Production
PSD	Process Shut Down
PSI	Pressure Safety Indicator
PT	Pressure Transmitter
RBF	Radial Basis Function
RNN	Recurrent Neural Network
SR	Sulphur Removal
SVM	Support Vector Machine
TI	Temperature Indicator
TSI	Temperature Safety Indicator
USD	Unit Shut Down
VARIMA	Vector Auto-Regressive Integrated Moving Average
VXI	Vibration in X-axis
VYI	Vibration in Y-axis
WI	Water Injection System

Summary

It is vital to maintain the stability and longevity of mechanical systems in the crucial field of vibration monitoring. The thesis introduces three sets of algorithms specifically tailored to fulfill the function to detect or predict vibration faults in the Water Injection Systems of Floating Production Storage and Offloading Unit (FPSO). These methodologies include PCA-based Prognosis, LSTM + One-class SVM, and VARIMA + One-class SVM. Notably, the LSTM + One-class SVM algorithm exhibits superior detective performance and robust resistance to data fluctuations, surpassing the other two approaches. Furthermore, it becomes evident that applying Principal Component Analysis for dimensionality reduction can adversely affect the discerning abilities inherent to both LSTM- and VARIMA-related algorithms.

In the context of predictive functionality, given the absence of definitive indicators, the development of predictive models remains contingent upon the existence of robust detective models. Both LSTM- and VARIMA-related algorithms (excluding PCA) demonstrate their efficacy in fulfilling this prerequisite. While the performance of PCA-based Prognosis continues to lag behind, its distinctive capacity to delve into the intricate patterns of equipment operational states hints at the possibility of unearthing richer insights compared to the aforementioned techniques. Consequently, the potential of PCA-based Prognosis to evolve into a viable predictive model should not be underestimated.

Contents

1 Introduction

Structural failure in engineering operations, particularly the failure of key devices, poses significant risks to operational efficiency and the safety of personnel. Such failures can result in substantial financial losses and create hazardous situations for workers. To mitigate these risks, engineering companies worldwide prioritize the monitoring of structural integrity and the early detection of abnormal behaviors. However, relying solely on human inspection for complex engineering systems is neither time-efficient nor labor-efficient, necessitating the development of advanced tools and strategies to enhance monitoring processes [1, 2].

Following such trends of digitalization, the Operational Intelligence and Performance Optimization Center (OIPOC) team of SBM Offshore N.V. focuses on development of solutions for engineering issues based on data science and computer science techniques, and it is one of its tasks to make analysis on vibration failure for systems installed on Floating, Production, Storage and Offloading (FPSO) unit. A Floating Production Storage and Offloading (FPSO) unit is a type of floating vessel used in offshore oil and gas production. It is designed to process and store hydrocarbons extracted from subsea wells, and then offload the processed oil or gas to shuttle tankers for transportation. This type of vessel consists of large amounts of rotating devices that suffer from long-term vibration, which can significantly increase the extent of structural fatigue of the whole structure.

In engineering, vibration monitoring is crucial for ensuring the reliability and safety of machinery and structures. The research on vibration monitoring can be divided into two main categories: Prediction and Detection. In the early stages, from the 1950s to the 1970s, vibration fault prediction relied heavily on vibration sensors. By monitoring and analyzing the signals generated during operation, experts could foresee potential issues. During this period, research focused on signal acquisition and processing techniques, including filtering and spectral analysis [3]. With the rapid development of computer technology in the 1980s and 1990s, digital signal processing techniques became crucial for vibration fault prediction. Notably, methods such as the Fast Fourier Transform (FFT) and wavelet transform emerged, enabling more sophisticated and precise analysis of vibration signals [4, 5]. These advancements allowed for better detection and diagnosis of machinery faults, improving maintenance practices and reducing downtime.

In the current era, machine learning and artificial intelligence have revolutionized vibration fault prediction. Advanced algorithms such as Support Vector Machines (SVM), neural networks, and deep learning offer unprecedented accuracy in analyzing and predicting vibration signals. These technologies, coupled with multi-sensor data fusion and big data analytics, enhance both the precision and timeliness of predictions [6, 7].

Detection methods have also evolved significantly. Before the 1950s, vibration faults were primarily detected through visual inspections, effective for simple structures but inefficient for complex machinery. The analytical phase began in the early 1960s with the advent of systematic vibration testing technologies. By examining the relationship between vibration and various factors, experts could preliminarily analyze faults, eliminating unrelated factors and narrowing down the fault suspicion range [8].

Post-1960s, as vibration testing technology evolved, a more structured approach to fault diagnosis emerged. Researchers delved deeper into vibration phenomena, characteristics, and mechanisms. Fault sources were systematically described based on vibration literature and compared with actual case data, allowing for precise analysis and diagnosis of vibration faults. This method of diagnostic thinking has been widely adopted and remains a cornerstone in the field [9].

Principal Component Analysis (PCA) has emerged as a powerful tool in data science for the reduction of data dimensionality. PCA transforms large datasets into smaller sets of new variables called principal components, which retain most of the original data's information [10]. This transformation simplifies the dataset while preserving its essential characteristics, making it easier to analyze and interpret. PCA has been widely applied in fault diagnosis to identify key factors contributing to structural anomalies [11]. By reducing the dimensionality of the data, PCA facilitates more effective and efficient time-series analysis, enabling the timely detection of abnormal events.

Listed below are the main research question this research focuses on with three sub-questions following by:

Main Question: How to detect the abnormal events of vibration from data of water injection systems of FPSOs using Principal Component Analysis?

Sub-Question 1: How to effectively reduce dimensionality of the datasets of the target problem using Principal Component Analysis?

Sub-Question 2: How to detect the abnormal events using the principal components of the datasets?

Sub-Question 3: What are the criteria for determining the optimal detection method?

The main question addresses the overarching goal of our research: leveraging strategies for effective vibration fault detection in Water Injection Systems of FPSOs. The sub-questions break down this goal into manageable components, focusing on data dimensionality reduction, the application of PCA in fault detection, and establishing criteria for optimal methods.

Sub-Question 1 deals with the challenges of handling high-dimensional data typical for vibration monitoring and explores PCA as a technique to simplify this data while retaining critical information.

Sub-Question 2 investigates how the transformed data (principal components) can be utilized for accurate detection of vibration faults, ensuring timely intervention.

Sub-Question 3 aims to define the criteria for evaluating the effectiveness and efficiency of the detection methods, ensuring they meet operational standards.

All in all, this research is developed in the basis of the research tasks of the OIPOC team and focuses on developing a real-time program for detecting vibration faults with and/or without PCA. Utilizing advanced data science techniques, we aim to monitor abnormal vibration events in the water injection systems. Our objective is to create a program capable of timely detection of abnormal vibration events and provide corresponding intervention measures to ensure the reliability and safety of FPSO operations. Through this study, we aspire to contribute to the ongoing advancement of structural health monitoring in complex engineering environments.

2

System Introduction and Data Acquisition

2.1. Incident Investigation

Based on the Process Stability Report of SBM Offshore, incidents categorized as Unit Shut Down (USD) for gas-related equipment (indicated as USD G) and water-related equipment (indicated as USD W) installed on 13 FPSOs, including CDM/CDI/CDS, account for the majority of incidents recorded. As depicted in Table 2.1, from the year 2020 to 2023, abnormal events of the USD G and USD W types constitute over 80 percent of all incidents logged, with both categories generally maintaining an equal share. Thus, it becomes apparent that delving deeper into the malfunction analysis of water- and gas-related equipments is of paramount importance.

	Number of Trips				
Year	USD G	USD W	PSD	ESD	Total
2023	222	159	50	25	456
2022	183	126	33	25	367
2021	107	116	35	20	278
2020	77	99	12	21	209
Total	589	500	130	91	1310
Ratio%	44.96	38.17	9.92	6.95	

Table 2.1: Trip Count of Recorded Incidents from 2020 to 2023

Among all recorded USD W/G type events from 2020 to 2023 (shown in Figure 2.1), the Water Injection System (WI) has contributed the highest number of abnormal occurrences (195), accounting for 17.91% of the total. This ratio is nearly double that of the Main Gas Compressor (MGC) with 99 events (9.09%) and the Gas Turbine Generator (GTG) with 83 events (7.62%), which follow in succession.

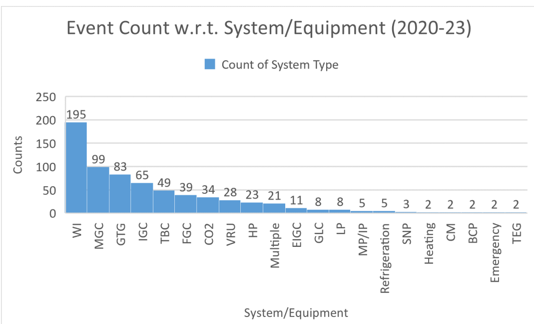

Figure 2.1: USD W and USD G Incident Counts 2020-2023

Additionally, as shown in Figures 2.2 and 2.3, which display the trip counts of USD W and USD G type events from 2019 to May of 2024 respectively, three FPSOs (CDI, CDM and CDS) generally face the most significant number of events. It is essential to highlight that the three FPSOs in question possess a shared framework for their Water Injection Systems, which facilitates the analysis of these systems' performance. Same frameworks imply that the datasets from these FPSOs have identical column headers (or features), enabling the development and testing of generalized code for diagnosing Water Injection System failures. By leveraging this commonality, we can streamline our analytical process and improve the reliability of our results, as the code can be applied across multiple FPSOs without the need for FPSO-specific adjustments. This makes CDI/CDM/CDS a good set of FPSOs for sampling one specific type of abnormal events, training models and testing our results.

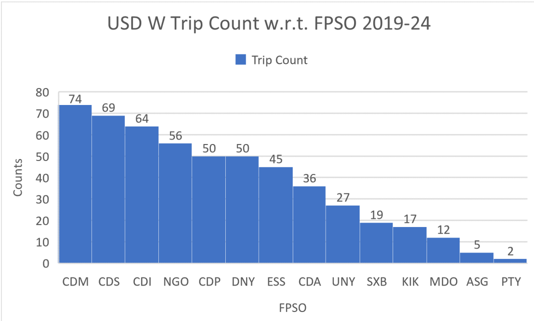

Figure 2.2: USD W Incident Counts from 2019 to May 2024

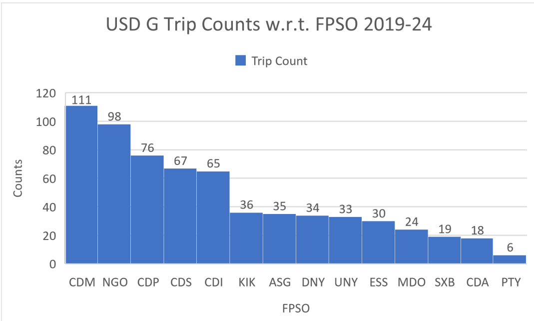

Figure 2.3: USD G Incident Counts from 2019 to May 2024

From 2022 to 2023, among all recorded abnormal events directly associated with water injection systems, vibration issues comprised the majority of anomaly types, as depicted in Figure 2.4. This finding is not surprising, considering that the system encompasses numerous rotating equipment, such as centrifugal compressors and pumps, which experience continuous vibration during their operation. A more comprehensive overview of the Water Injection System is provided in section 2.2.

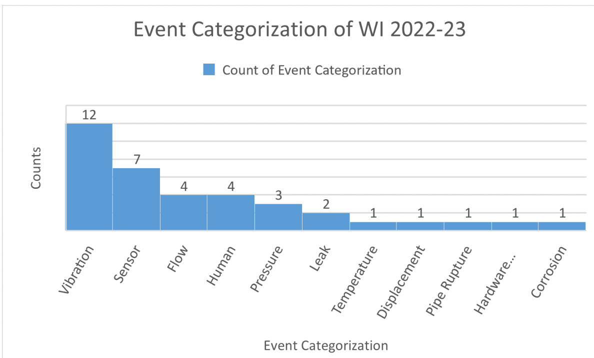

Figure 2.4: Incidents Counts of Water Injection System form 2020 to 2023 [CDI/CDM/CDS]

2.2. Introduction to Water Injection System

As shown in Figure 2.5, as one of the most vital modules of FPSOs' topsides systems, Water Injection System (WI) works together with other equipments, such as Main Gas Compression System, to maintain various production procedures.

WI consists of devices of seawater treatment and water injection. After seawater is transformed into filtered, chemically treated, low sulphate, deaerated water stream through Seawater Treatment System, it is driven by the embeded pumps (HP Feed Pump, Water Injection Pump, etc.) to water injection headers where the stream is distributed to designated subsea water injection and so on.

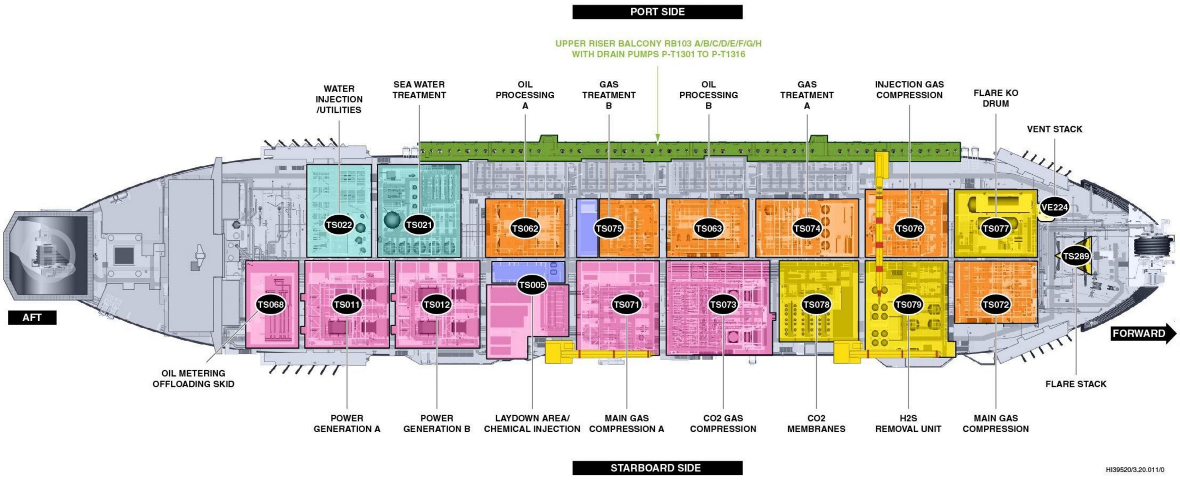

Figure 2.5: Layout of Topsides Systems

More detailedly, as shown in Appendix A, several principal equipments function together to fulfill the work of Water Injection System:

1) Seawater Basket Filters S-T2202A/B/C
2) Media Filters S-T2671A/B/C/D/E
3) De-aeration Column V-T2601 and Vacuum Package A-T2660
4) HP-Feed Pump P-T2631A/B
5) Guard Filters S-T2631A/B/C/D
6) SRP Membrane Units A-2631A/B and CIP Package A-T2632
7) Water Injection Pumps P-T2611/21
8) Water Injection Chemical Injection Package A-T2850

Seawater from a depth of approximately 50 meters is filtered through coarse and fine filters, removing particles larger than 5 microns. The filtered seawater undergoes vacuum de-aeration to remove dissolved oxygen to less than 50 parts per billion (ppb), with oxygen scavenger added to chemically reduce the remaining oxygen to a maximum of 10 ppb and eliminate residual chlorine. The de-aerated seawater is then boosted to the required pressure for the Sulphate Removal Unit membranes. After passing through parallel SRUs, the injection water achieves the specified sulfate level of 40 parts per million (ppm). Finally, Water Injection Pumps supply the de-aerated, low-sulfate injection water at the necessary pressure and flow for well injection.

2.3. Data Acquisition

This research commences by thoroughly examining the Process Stability Report of SBM Offshore and files from SRS database, which act as the repository for the historical data of FPSOs and their respective faulty events. This comprehensive analysis allows us to delve into the intricacies of the past incidents, providing valuable insights that can guide our subsequent investigations.

Within the vast array of recorded incidents, our attention is focused on those specifically associated with the vibration of Water Injection system. The system, as introduced above, plays a crucial role in maintaining the smooth operation of FPSOs, and any malfunction or degradation can have a significant impact on the overall efficiency and productivity of the platform. By isolating incidents related to vibration of Water Injection system, we aim to gain a deeper understanding of the root causes and contributing factors behind these events.

According to the Process Stability Report, we have effectively pinpointed 10 malfunction events between 2022 and 2023 that are directly linked to the Water Injection Systems of three FPSOs: CDI, CDS, and CDM. As previously discussed in the preceding section, these three FPSOs, although separate entities, exhibit similarities in their respective Water Injection systems. Parts of the event details are presented in Table 2.2.

FPSO	Year	Month	Day	Type	Failure Category	Detailed Equipment
CDM	2023	August	9	USD W	Mechanical Failure	Water Injection Pump
CDS	2023	May	31	USD W	Mechanical Failure	HP Feed Pimp
CDI	2023	February	27	USD W	Instrument Failure	HP Feed Pimp
CDM	2023	July	13	USD W	Instrument Failure	HP Feed Pimp
CDM	2022	February	19	USD W	Process Conditions	HP Feed Pimp
CDI	2022	November	25	USD W	Other	HP Feed Pimp
CDS	2022	May	27	USD W	Mechanical Failure	Water Injection Pump
CDS	2022	April	6	USD W	Instrument Failure	HP Feed Pimp
CDM	2022	June	11	USD W	Instrument Failure	Water Injection Pump
CDI	2022	August	12	USD W	Instrument Failure	HP Feed Pimp

Table 2.2: Recorded Events of Vibration From 2022 to 2023

Additionally, we have identified another 12 vibration-related events of Water Injection System ranging from 2019 to 2021 from SRS database. These events also belong to CDI, CDM and CDS. Parts of event information is listed in Table 2.3. However, due to the issues that the data among some of these events suffer from large-scale missing values, bring great barriers to identify abnormality and collect sufficient data to train and test model, only 12 of the events listed in Table 2.3 and Table 2.2 can be used in this research.

Reference	Incident Type	Detailed Equipment
CDI20190131-001	EF PT	HP Feed Pump
CDI20190512-001	PT	HP Feed Pump
CDI20190513-002	EF PT	Water Injection Pump
CDI20201028-001	EF PT	Water Injection Pump
CDI20210124-001	EF PT	Water Injection Pump
CDI20210212-001	EF PT	Water Injection Pump
CDM20190207-001	EF PT	HP Feed Pump
CDM20190306-001	EF	HP Feed Pump
CDM20200913-003	EF PT	Water Injection Pump
CDM20201124-001	EF PT	HP Feed Pump
CDM20210312-001	EF PT	Water Injection Pump
CDM20210315-001	EF PT	Water Injection Pump

Table 2.3: Recorded Events of Vibration From 2019 to 2021

As we can see from the records of vibration-related events mentioned above, Water Injection Pump and HP Feed Pump contribute to all vibration-related incidents of Water Injection System.

After filtering out those events with missing data, 6 events are finally picked up (4 for Water Injection Pump, 2 for HP Feed Pump) from the remaining datasets of 12 events. We extract the complete faulty condition data for each of the 6 selected events with respect to equipment, meticulously examining every second of their operational history. This comprehensive approach allows us to capture even the most subtle nuances and fluctuations in the WI systems' performance during these critical periods. In addition to the faulty condition data, we also gather a sequential record of the healthy condition that is close to each fault in history.

By analyzing both the faulty and healthy condition data second by second, we can gain a deeper understanding of the factors that contribute to vibration incidents of Water Injection system. This level of granularity is essential for constructing robust detective/predictive models that can accurately identify impending faults and provide operators with sufficient time to take corrective action.

Herein, we present several lines of representative data extracted from one dataset in Table 2.4, which serve as excellent examples for showcasing the structure and composition of the generic data. These examples provide a glimpse into the information we have compiled, offering valuable insights into the various facets of the functionality and performance of Water Injection system.

As evident in the sample dataset, we have compiled a comprehensive set of measurements encompassing various aspects of the Water Injection Systems of the three FPSOs. The dataset features a column of time stamps, which serves as a reference point for each observation, and a column of labels denoting the operational status of the WI systems. In this research, we have established a clear and intuitive labeling scheme: a value of 0 corresponds to a 'Healthy' state, while a value of 1 represents a 'Faulty' state. Those labeled states are determined by correlating the data in the datasets with descriptions of the corresponding incident records in databases.

Date	PM-T2611/kW	PM-T2621/kW	...	T26-PI-1972/barg	T26-LI-1973/%	label
10/28/2020	5065.137	3384.558	...	13.78099	73.11741	0
10/28/2020	5065.137	3384.558	...	13.72122	72.97891	0
10/28/2020	5065.137	3384.558	...	13.66145	72.84042	1
10/28/2020	5065.137	3384.558	...	13.60168	72.70192	1

Table 2.4: Several Exemplary Samples Extracted from One Dataset

Complementing the two columns are numerous others that encapsulate a wide array of features relevant to the WI systems' performance. In total, we have amassed 64 features for Water Injection Pump and 25 for HP Feed Pump, each feature offering valuable insights into the system's operational health. These features/tags are derived from the extensive database maintained by SBM Offshore, a repository that meticulously logs every aspect of the FPSOs' operations using a system of tag numbers. These features/tags simplify the data retrieval process, allowing researchers to quickly locate and analyze specific variables of interest. Since this research focus on vibration-related problems, we further processed the data obtained. Appendix B provides a detailed description of each operational feature along with its corresponding tag number. This comprehensive overview offers a snapshot of the rich information available in our datasets, laying the groundwork for the development of predictive models and diagnostic tools.

While gathering vibration-related incidents, we classify these events according to the specific equipment that causes the issue. The classification outcomes are presented in the last columns of Table 2.2 and 2.3. These tables unequivocally demonstrate that Water Injection Pumps and HP Feed Pumps are significant contributors to the recorded vibration-related anomalies in the Water Injection System. To enhance the precision of our further analysis, this study will concentrate on both types of equipment independently. The tags of the two subsystems are listed in Appendix C and Appendix E. Afterwards, through pre-experiments, we found that features that are not related to vibration, such as temperature and flow rate, can introduce a lot of fault interference from other equipment. However, since those vibration indicators only record the vibration of the equipment they are in, and the different devices are basically physically independent, the values of the vibration indicators can be more independent than other indicators.. Thus only VI-, VXI- and VYI- types of tags/features are kept within the datasets as shown in Appendix D and Appendix F, while the others are removed.

Additionally, as each of both equipments comprises two sets of sub-equipment, designated as A and B, which typically exhibit identical structures and functions, they are expected to operate simultaneously. But under special circumstances such as maintenance, sub-equipment A and B might show different operation patterns, thus they should be regarded as separate devices though mechanically similar. Consequently, our analysis is meticulously carried out with a focus on each individual sub-equipment.

3
Theory

The advent of various computational principles has led to the evolution of data processing models from simple to complex, as exemplified by Principal Component Analysis (PCA), Vector Auto-Regressive Integrated Moving Average (VARIMA), and Long Short-Term Memory (LSTM). These models are distinct in their capacity to extract features from raw data, demonstrating a sequential enhancement in their ability to capture increasingly complex patterns. However, this increase in model complexity also corresponds to a higher time cost for training. In light of these considerations, this study considers utilizing PCA to decrease computational workload and extract statistical characteristics. However, since PCA is merely able to capture little nonlinear information from data, another two more advanced regressors, namely VARIMA and LSTM, are applied as the foundational techniques for vibration fault detection.

Following the identification of preliminary health characteristics, the next step involves establishing thresholds for false alarms. For PCA, we establish thresholds directly using empirical healthy data. In contrast, for VARIMA and LSTM, which incorporate multiple indicators, we employ One-class Support Vector Machine (One-class SVM) to assist in automating this process. By comparing and assessing the outcomes generated by these three methodologies, our aim is to discern the balance between model complexity, training time, and detection accuracy.

This chapter offers a generic description of the theories used in this research, covering principles of PCA, VARIMA, LSTM and One-class Support Vector Machine.

3.1. Principal Component Analysis

Principal Component Analysis (PCA) is a powerful technique widely employed in data analysis, offering insights into the intricate relationships among various features. Beyond its analytical prowess, PCA serves as an invaluable tool for dimensionality reduction, a process crucial for handling high-dimensional datasets efficiently.

At its core, PCA aims to identify the principal components within a dataset, which are orthogonal vectors that capture the maximum variance present in the data. In this way, PCA helps to discern underlying patterns and structures that might not be immediately apparent in the raw data. The obtained principal components not only simplify the original datasets but also retain much of the essential information contained within them.

Assume that there is a datasets X, which consists of p samples.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_p \end{bmatrix} \tag{3.1}$$

Assume there are n features for the p-dimensional observation space, and we can acquire the observation matrix of Equation (3.1). (Both n, p are finite positive integers)

$$X = \begin{bmatrix} x_{11} & x_{12} & ... & x_{1n} \\ x_{21} & x_{22} & ... & x_{2n} \\ ... & ... & ... & ... \\ x_{p1} & x_{p2} & ... & x_{pn} \end{bmatrix} \tag{3.2}$$

In this way, we can estimate the general variance-covariance matrix of with that calculated based on the data in the observation matrix (3.2), indicated as matrix S.

$$S = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{bmatrix} \tag{3.3}$$

where

$$a_{ij} = \frac{1}{p} \times \sum_{k=1}^{p} (x_{ik} - \tilde{x}_i)(x_{jk} - \tilde{x}_j) \qquad (i, j = 1, 2, ..., p)$$

Besides, the mean value vector $\tilde{x}$ can be regarded as an estimator of general vector U. The fundamental aim of PCA is to find a new matrix V, which is smaller that X in dimensions and contains linear combinations with maximum variance. Such new matrix V, also known as principal component matrix, share the quantitative relationship written below with X and S:

$$V = A^T X \tag{3.4}$$

where
V: principal component matrix,
A: a matrix of orthonormal eigenvectors of matrix S, also knwon as eigenbasis,
X: original observation matrix.

Equation (3.4) can be solved after getting results from the following deterministic equation:

$$|S - lI| = 0 \tag{3.5}$$

where
l: the eigenvalues of matrix S,
I: unit matrix of size $(n \times n)$.

We can get n values of l after calculation, indicated as l_1, l_2, ... , l_n. As for each eigenvalue, there is a corresponding orthonormal column eigenvector A_i (i = 1, 2, ..., n). Assume that all l values are arranged from large to small numerically.

$$l_1 \geq l_2 \geq ... \geq l_p \tag{3.6}$$

Normalize the l values, and indicate them as $C_i \;\; (i = 1, 2, ..., n)$. Those C_i values can be regarded as weights in the linear combinations, which in other words means that the larger the C_i value, the more information the corresponding V column vector contains. For example, the leftmost column vector V_1 in matrix V corresponds to C_1 and is termed as the *first principal component*.

$$C_i = \frac{l_i}{l_1 + l_2 + ... + l_p} \tag{3.7}$$

Another important step of PCA is to filter key principal components and erase the others. Usually, we can determine the number of key principal components through the cumulative variance contribution rate (Cumulative Variance Explained). The cumulative variance contribution rate represents the percentage of the total variance explained by the first k principal components to the total variance of the original data. A commonly used selection criterion is that the cumulative variance contribution rate reaches more than 95%. The setup of threshold value of contribution ratios is based on the task requirements and subjective judgement. After that, researchers shall conduct interpretation on each principal component left according to the characteristics of variables included, to determine the generic element it represents.

Generally speaking, PCA shows the advantages of significant dimensionality reduction and computational efficiency. By reducing the dimensionality of the original data set, PCA facilitates the visualization and processing of data while reducing the impact of redundant information. In addition, PCA can also remove noise by eigenvalue decomposition to improve the accuracy and reliability of data. For large-scale data computing, PCA can convert the computation process into the computation of a small number of eigenvectors, thereby greatly improving the computational efficiency.

However, as mentioned at the beginning of chapter 3, PCA is lack of ability to analyze nonlinear data. Besides of such issue, PCA also has some other drawbacks and limitations [12]. First of all, PCA is sensitive to outliers, and the presence of outliers may lead to the deviation of the extracted principal components from the real situation. Second, the PCA assumes that the data conforms to a Gaussian distribution, and if the actual data distribution does not conform to this assumption, the analysis results may be inaccurate. In addition, PCA-extracted principal components are often difficult to interpret their specific meanings, requiring additional analytical and interpretive work to draw conclusions. Finally, the application of PCA is limited by the sample size and the number of variables, and insufficient sample size or too many variables may lead to the extraction of principal components that are not representative.

3.2. Vector Auto-Regressive Integrated Moving Average

Prediction on future development of time-series data is one important part of predictive maintenance procedure, which helps inspectors and operators to foresee the possible abnormal circumstances of devices.

Multiple data prediction methods have been proposed to fulfill the task, including AR/MA-based statistical tools. The concept of Auto-Regression was first proposed by George Yule [13] in 1927. As one of the earliest time prediction methods, Auto-Regression (AR) model describes the auto-correlation of time series by expressing the observation value at the current moment as a linear combination of the observation values at several past moments.

$$y_{(t)} = c + \phi_1 y_{(t-1)} + ... + \phi_p y_{(1)} + \epsilon_{(t)} \tag{3.8}$$

where
$y_{(t)}$: observation value at time t,
c : constant term,
ϕ_1, ... , ϕ_p: coefficients of auto-regression,
p: order of auto-regression,
$\epsilon_{(t)}$: sequence of white noise.

Norbert Wiener [14] then presented another approach of time-series data prediction in 1940, namely Moving Average (MA). Different from the AR model, the MA model describes the dependence of the time series by expressing the observation value at the current moment as a linear combination of error terms at several past moments.

$$y_{(t)} = c + \epsilon_{(t)} + \theta_1 \times \epsilon_{(t-1)} + \theta_2 \times \epsilon_{(t-2)} + ... + \theta_q \times \epsilon_{(t-q)} \tag{3.9}$$

where
$y_{(t)}$: observation value at time t,
c : constant term,
θ_1, θ_2 ... , θ_q: coefficients of moving average,
q: order of moving average,
$\epsilon_{(t)}$: sequence of white noise.

Peter Whittle [15] combined the two methods together (Equation 3.8, Equation 3.9) and proposed the concept of Auto-Regressive Moving Average (ARMA). The ARMA model can simultaneously capture the auto-correlation and dependence of time series.

$$y_{(t)} = c + \phi_1 y_{(t-1)} + ... + \phi_p y_{(1)} + \epsilon_{(t)} + \theta_1 \times \epsilon_{(t-1)} + \theta_2 \times \epsilon_{(t-2)} + ... + \theta_q \times \epsilon_{(t-q)} \tag{3.10}$$

where $y_{(t)}$, c, {θ_1, θ_2 ... , θ_q}, q, p, $\epsilon_{(t)}$ and {ϕ_1, ... , ϕ_p} share the same numerical meanings with those mentioned above.

However, AR, MA and ARMA models usually require the data to be stationary, i.e. the statistical characteristics of each order (such as mean, variance, covariance...) of a set of time series data do not change with time. This results in algorithms such as AR/MA being very limited in their prediction effects on non-stationary data. At the same time, the amount of information these algorithms can capture in processing multi-variable time series data is also relatively limited. As one of the extended forms of the ARMA model, the ARIMA model was proposed to solve the above problems. This model handles non-stationary time series by introducing difference operations.

$$(1-B)^d \times y_{(t)} = c + \phi_1 y_{(t-1)} + ... + \phi_p y_{(1)} + \epsilon_{(t)} + \theta_1 \times \epsilon_{(t-1)} + \theta_2 \times \epsilon_{(t-2)} + ... + \theta_q \times \epsilon_{(t-q)} \tag{3.11}$$

where B is the lag operator, d is the difference order, while $y_{(t)}$, c, {θ_1, θ_2 ... , θ_q}, q, p, $\epsilon_{(t)}$ and {ϕ_1, ... , ϕ_p} share the same numerical meanings with those mentioned above.

The VARIMA model is a multidimensional generalization of the ARIMA model and is used to analyze multidimensional time series.

$$(1-B)^d \times Y_{(t)} = c + \phi_1 Y_{(t-1)} + ... + \phi_p Y_{(1)} + \epsilon_{(t)} + \theta_1 \times \epsilon_{(t-1)} + \theta_2 \times \epsilon_{(t-2)} + ... + \theta_q \times \epsilon_{(t-q)} \tag{3.12}$$

where B is the lag operator, d is the difference order, $Y_{(t)}$ is multidimensional time series, while c, {θ_1, θ_2 ... , θ_q}, q, p, $\epsilon_{(t)}$ and {ϕ_1, ... , ϕ_p} share the same numerical meanings with those mentioned above.

The flexibility of the VARIMA model is one of its greatest advantages. Due to its ability to adapt to various types of linear time series data, whether stationary or non-stationary, the VARIMA model has been widely used in finance, economics, meteorology, ecology and other fields. This versatility makes the VARIMA model more adaptable when facing complex problems. At the same time, the VARIMA model has excellent predictive capabilities. By capturing key characteristics such as trends, seasonality and periodicity in time series, the VARIMA model can provide decision makers with valuable information about future developments. In addition, the VARIMA model also has better interpretability. Its parameters have clear statistical significance, making it easier for data scientists to understand the dynamic relationships captured by the model.

The VARIMA model also has some limitations [16]. First, it has high data requirements and requires sufficient and high-quality time series data. Secondly, the computational complexity of VARIMA models is relatively high, especially when dealing with large-scale data sets. This can result in longer computation times and greater resource consumption. In addition, the problem of model selection is also a challenge faced by the VARIMA model. The VARIMA model involves the selection of multiple parameters, such as autoregressive terms, moving average terms, and lag orders. Choosing appropriate parameters is critical to a model's predictive performance, but this process may require multiple trials and validations. Finally, although the VARIMA model is able to capture a certain degree of nonlinear dynamics, its core is still built based on linear relationships. Therefore, in some cases it may not adequately capture all nonlinear relationships in time series.

3.3. Long Short-term Memory

Long Short-Term Memory (LSTM) is a specialized type of recurrent neural network (RNN) designed to address the challenges of modeling long sequences of data. LSTM tackles the vanishing gradient problem inherent in traditional RNNs, which occurs when learning long-term dependencies. This design enables LSTM to retain important past information while avoiding the overwrite issues that plague standard RNNs. In other words, LSTM exhibits superior robustness when dealing with long lists, a feature that sets it apart from traditional RNN networks. Figure 3.1 demonstrates the macroscopic schematics to explain how one RNN model works.

As shown in Figure 3.2, LSTM achieves this by incorporating gated mechanisms known as forget gates, input gates, and output gates. These gates regulate the flow of information and the updating of memory, allowing the network to selectively remember or forget information over time.

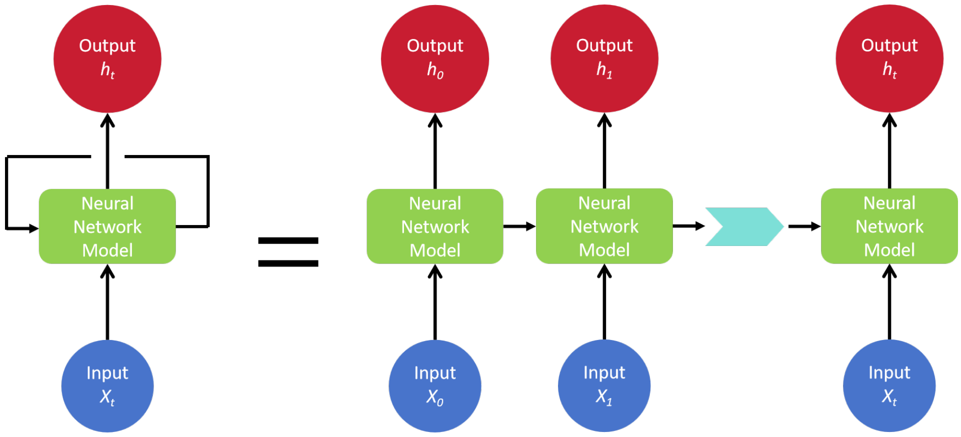

Figure 3.1: Framework of LSTM [17]

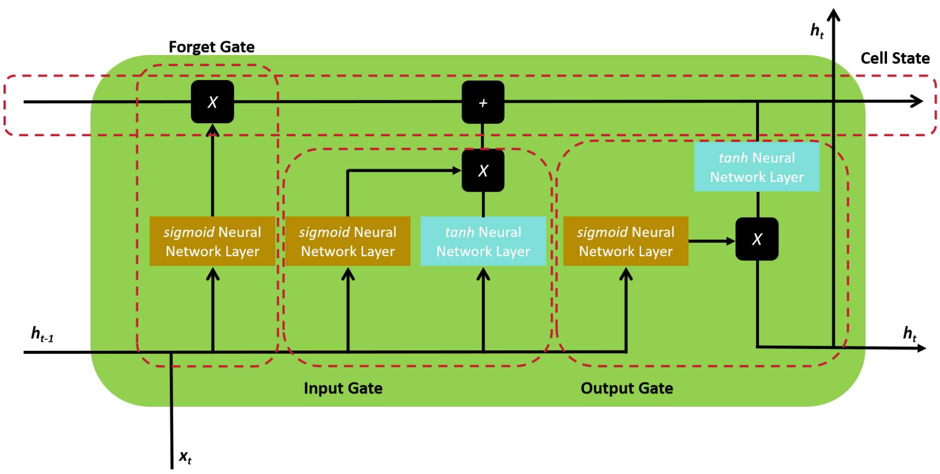

Figure 3.2: Framework of LSTM Gate Mechanism [18]

The forget gate serves a crucial role in LSTM by determining what information from the previous time step should be retained or forgotten in the current time step. It takes as input the previous cell state and the current input, applies a sigmoid activation function, and outputs a vector of values between 0 and 1 representing the degree to which each element of the cell state should be retained. A value close to 0 indicates that the corresponding memory should be forgotten, while a value close to 1 indicates that it should be retained.

Next, the input gate decides which new information should be stored in the cell state. It consists of a sigmoid activation function and a tanh activation function. The sigmoid function determines the relevance of each element in the input, while the tanh function generates a new candidate value. These values are then combined to produce an update to the cell state.

Finally, the output gate determines the final output of the LSTM cell. It regulates which parts of the cell state should be passed to the output. The gate consists of a sigmoid activation function to determine which parts of the cell state to output and a tanh activation function to scale the output. The output gate then combines these two outputs to produce the final output of the LSTM cell.

The cell state in a LSTM network serves as the memory of the network, allowing information to flow across different time steps while selectively retaining or discarding information. It acts as a conveyor belt that carries information throughout the sequence, and its state can be updated through various operations involving the forget gate, input gate, and output gate. The cell state retains information over long periods, enabling the LSTM to capture dependencies in sequential data.

LSTM networks offer several advantages in sequential data processing, excelling in capturing long-term dependencies. With their unique gated mechanisms, LSTM effectively mitigates the vanishing gradient problem commonly encountered in traditional recurrent neural networks, enabling robust learning even in the presence of lengthy sequences. This flexibility allows LSTM to adapt to variable sequence lengths without requiring predefined input dimensions, making it versatile across diverse applications. However, LSTM's intricate architecture and numerous parameters contribute to high computational costs, potentially leading to overfitting in resource-limited settings. Furthermore, the complexity of LSTM models poses challenges in interpretation and explanation. Nevertheless, LSTM remains widely utilized in natural language processing, time series analysis, and speech recognition domains due to its unparalleled ability to learn and model long-term dependencies in sequential data.

3.4. One-class Support Vector Machine

One-class Support Vector Machine (One-class SVM) [19] is a machine learning algorithm used for anomaly detection, particularly when only one class of data is available for training. Unlike traditional SVM, which is a supervised learning algorithm used for classification, One-class SVM is an unsupervised learning algorithm focused on learning the distribution of normal data points and identifying deviations from this distribution. This kind of model feature enables the one-class SVM to be directly used for model training and testing after the feature extraction task of the regressor is completed, without the need to introduce additional training sets or further feature engineering to original datasets.

At the core of One-class SVM lies the concept of constructing a hyperplane in a high-dimensional feature space. This hyperplane serves as the decision boundary, separating normal data points from potential outliers. Mathematically, the hyperplane equation is represented as $\omega^T \cdot \phi_{(x)} + b = 0$, where ω is the normal vector, $\phi_{(x)}$ is the feature function, and b is the bias term. The goal is to optimize ω and b such that they define a hyperplane that maximizes the separation margin around the normal data points.

To achieve this, One-class SVM formulates an optimization problem aimed at minimizing the empirical risk while maximizing the margin around the normal data points. The optimization problem incorporates slack variables, allowing for a certain degree of flexibility in accommodating normal data points that lie on the wrong side of the hyperplane. Additionally, the parameter ν is introduced to control the trade-off between capturing the normal data distribution and allowing for outliers. By adjusting ν, the algorithm can adapt to different data distributions and anomaly detection requirements.

A crucial aspect of One-class SVM is its ability to handle nonlinearities in the data through the kernel trick. The kernel trick allows the algorithm to implicitly map the input data into a higher-dimensional feature space, where linear separation becomes possible. Common kernel functions used include the radial basis function (RBF) and polynomial kernels. By mapping the data into higher-dimensional spaces, One-class SVM can effectively capture complex relationships in the data and learn nonlinear decision boundaries, enhancing its versatility and applicability to various real-world scenarios.

One-class SVM offers advantages including effectiveness with unbalanced data, robustness to high-dimensional data, and versatility with kernel functions, but it is sensitive to hyperparameters, has limited interpretability, and may struggle with complex data distributions [20]. Despite these limitations, One-class SVM remains a powerful tool for anomaly detection, particularly suitable for scenarios with skewed data distributions and where understanding the reasoning behind outlier classifications is less critical.

4 Methodology

In this chapter, two distinct methods for detecting abnormalities are presented: PCA-based Prognosis and VARIMA/LSTM + One-class SVM. The PCA-based Prognosis method focuses on leveraging principal component analysis (PCA) to extract eigenbases, which serve as representative vectors capturing the essential variability within the dataset. By analyzing the eigenbases, the method identifies thresholds indicative of shifts in correlation among features. This approach offers a systematic means to detect abnormalities by discerning significant deviations from the established correlations observed during normal operational states. Meanwhile, the VARIMA/LSTM + One-class SVM method combines various techniques to enhance anomaly detection. It begins by reducing the dimensionality of datasets through PCA, followed by the utilization of machine learning and recurrent neural network (RNN) models to extract characteristic patterns from healthy-state feature values. Subsequently, it scrutinizes abnormal deviations by comparing residuals between actual and predicted values, employing One-class SVM to discern anomalies from normal behavior. These approaches collectively provide comprehensive strategies for anomaly detection, addressing different aspects of data analysis and modeling to effectively identify deviations indicative of potential issues.

4.1. Indicator

In order to quantify the performance of algorithms, four numerical indicators are introduced, namely **Accuracy**, **Precision**, **Recall**, and **F1 Score**. These four measurements are essential metrics for evaluating classification models. **Accuracy** measures the proportion of correctly classified samples out of the total samples, calculated as

$$\textbf{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

where TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives, respectively. It is best suited for balanced datasets. **Precision** quantifies the proportion of true positive predictions among all positive predictions, given by

$$\textbf{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

, making it crucial when the cost of false positives is high, such as in spam detection. **Recall** indicates the proportion of actual positive samples that are correctly identified, calculated as

$$\textbf{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

, and is particularly important in scenarios where missing a positive case is costly, such as in disease screening. The **F1 Score** is the harmonic mean of Precision and Recall, defined as

$$\textbf{F1 Score} = \frac{2 \times \textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}$$

, and is useful for evaluating models on imbalanced datasets where a balance between **Precision** and **Recall** is needed.

4.2. Work Flow of Principal Component Analysis

Through the fault data of WI systems recorded in actual engineering applications, PCA technology can help us identify key principal components related to vibration faults. These principal components can clearly display the characteristic modes of vibration faults, thereby providing strong support for fault diagnosis and early warning. By monitoring the changes in these principal components in real time, we can promptly detect and warn potential vibration faults, and take appropriate measures to intervene and deal with them to ensure the stability and safety of the WI system.

4.2.1. KMO Test and Sphericity Test

Before extracting the principal components, it is an indispensable step to perform KMO (Kaiser-Meyer-Olkin) test and sphericity test on the datasets. These tests help evaluate the quality of the datasets and ensure that they meet the conditions for factor analysis, thereby improving the analysis effect and application value of PCA.

4.2.1.1 KMO Test

The KMO test is mainly used to measure the degree of correlation between variables in the datasets. Specifically, the statistic of the KMO test is obtained by calculating the average of the ratio of the partial correlation coefficient to the correlation coefficient of all pairs of variables. The partial correlation coefficient refers to the degree of correlation between two variables while controlling other variables. The correlation coefficient is the direct correlation between two variables. The calculation formula of KMO statistic is as follows:

$$KMO = \frac{\sum r_{ij}^2}{\sum (r_{ij}^2 + p_{ij}^2)} \tag{4.1}$$

where, r_{ij} represents the correlation coefficient between variable i and variable j, and p_{ij} represents the partial correlation coefficient between variable i and variable j when controlling other variables. The subscripts i and j represent different pairs of variables respectively.

KMO value range is between 0 and 1. The closer the value is to 1, the stronger the correlation between variables and the higher the structural validity of the data set. When the KMO value is greater than 0.5, the data set is generally considered suitable for factor analysis. Through the KMO test, we can initially determine whether the dataset meets the basic conditions for PCA.

4.2.1.2 Sphericity Test

The sphericity test (Bartlett's sphericity test) further verified the distribution characteristics of datasets. The null hypothesis (H_0) of this test is that the variables are independent of each other, that is, the data has a spherical distribution. The alternative hypothesis (H_1) is: the correlation matrix is not the identity matrix, that is, there is a certain correlation between variables.

The statistic (χ^2) of Bartlett's test of sphericity is calculated as follows:

$$\chi^2 = -[(n-1) \times ln(S) - \sum (ln(\lambda_i))] \tag{4.2}$$

where n is the number of samples, S is the determinant value of the sample covariance matrix, and λ_i is the eigenvalue of the correlation matrix.

According to the χ^2 distribution table, we can find the corresponding degrees of freedom (df), usually $\frac{(p-1)\times(p-2)}{2}$, where p is the number of variables. We can then calculate the P value for Bartlett's test of sphericity. If the P value of the sphericity test is less than 0.05, the null hypothesis is rejected, indicating that there is a significant correlation between the variables and the data is non-spherical. In this case, the data set is more suitable for factor analysis in order to reveal the underlying relationships between variables.

Through the double checks of KMO test and sphericity test, we can ensure that the data set has the structural validity and distribution characteristics required for PCA. This helps to improve the analytical accuracy of PCA and reveal the main components and potential patterns in the data set, thereby laying a solid foundation for subsequent data mining and analysis work.

4.2.2. PCA Execution

After successfully obtaining the datasets and verifying that they are suitable for PCA analysis, we officially enter the critical stage of data preprocessing and principal component extraction. This stage is critical to ensure the accuracy and reliability of the analysis results. We will take a series of measures to optimize the data structure, extract core information, and lay a solid foundation for subsequent principal component interpretation and predictive analysis.

First, we **clean and standardize the data in each dataset**. Our operations in this step include removing obvious outliers in healthy states, filling in missing values with mean healthy value, modifying inappropriate data types, etc. to ensure a clean and consistent data set. Standardization scales all variables to the same scale, eliminating the influence of different units and dimensions, making the data more suitable for PCA analysis. Commonly used data standardization methods include min-max standardization, Z-score standardization, and proportional standardization. The formulas for these methods are given below:

1) Min-Max Scaling. To perform a linear transformation on the original data so that the resulting value maps to [0, 1].

$$x' = \frac{x - min}{max - min} \tag{4.3}$$

where x' is the standardized data, x is the original data, min and max are the minimum and maximum values of the original data respectively.

2) Standard Scaling. To subtract the original data from its mean and dividing by the standard deviation results in a normal distribution with mean 0 and standard deviation 1.

$$x' = \frac{x - \mu}{\sigma} \tag{4.4}$$

where x' is the standardized data, x is the original data, μ is the mean of the original data, and σ is the standard deviation of the original data.

3) Robust Scaling. To scale the data using quartiles (1st quartile and 3rd quartile).

$$x' = \frac{x - Q_1}{Q_3 - Q_1} \tag{4.5}$$

where x' is the normalized data, x is the original data, Q_1 is the first quartile of the original data, and Q_3 is the third quartile of the original data. This method is robust to outliers.

Considering the raw data still contains outliers within healthy state, this thesis chooses the second standardization method since it is less susceptible to extreme values compared with the other methods, providing more stable feature scaling.

Next, we perform **principal component extraction** on the data. By solving for eigenvalues and eigenvectors, we obtain the principal components of the dataset that explain most of the variation in the data. We also determine the information contribution ratios of the principal components based on the magnitude of corresponding eigenvalues and select one or several principal components to represent the original data.

In short, in the data preprocessing and principal component extraction stage, we take a series of rigorous measures to decrease the dimensionality of original datasets and/or extract key information. This operation provides strong support for subsequent principal component interpretation and model training, helping us better understand and utilize the data.

4.3. Approach 1: PCA-based Prognosis

PCA-based Prognosis relies on detecting abnormal changes in correlations among the equipment within a system during abnormal events. As shown in fig 4.1, this method utilizes the eigenbasis of correlation matrices derived from datasets to unveil the principal correlations, thereby enabling the examination of the extent of deviation in the eigenbasis during normal conditions. By establishing prognostic thresholds based on these observations, early alarms can be triggered when deviations exceed predefined limits. This approach ensures timely intervention and maintenance, safeguarding the system's functionality and efficiency.

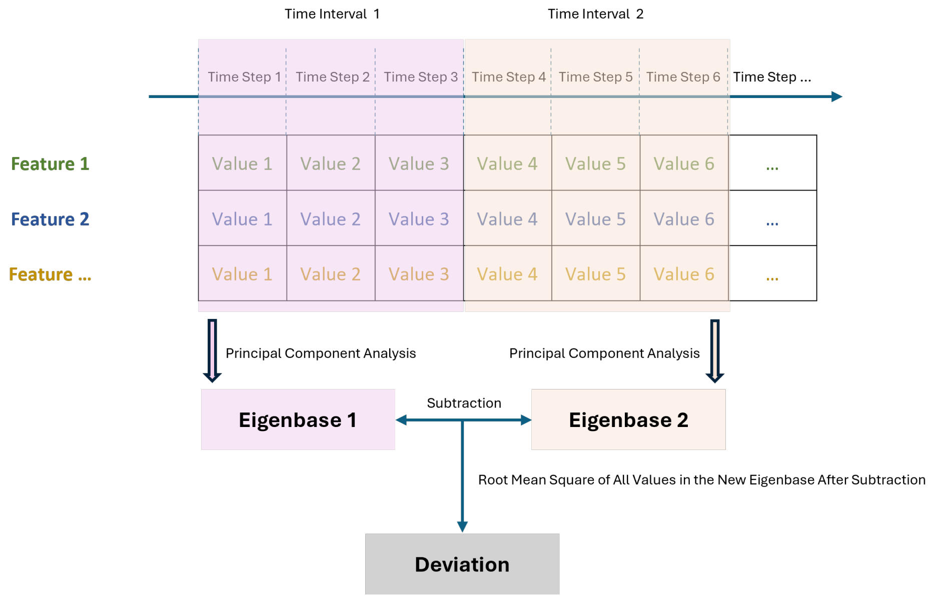

Figure 4.1: Schematics of PCA-based Prognosis for the first two consecutive time intervals. The whole multi-dimensional dataset is divided into many time intervals encapsulating equal number of time steps (3 steps shown in the figure in order to save space, but 60 steps enclosed in one set for this research actually).

4.3.1. Data Collection and Feature Engineering

We collect those datasets that contain only the time-series data during the healthy state of the interested equipments. Each dataset is first divided into certain number of time intervals each with 60 time steps.

Following data collection, we engage in **feature engineering** to prepare the data within each time interval for principal component analysis. This involves at least two crucial procedures: Firstly, eliminating features with constant values to prevent issues such as zero-value determinant during eigenvalue and eigenbasis calculation. Secondly, standardizing the values of each feature to mitigate bias concerns stemming from dimensionality discrepancies.

4.3.2. PCA and Deviation of Eigenbases

Subsequently, PCA is employed to **acquire the eigenvectors and eigenbases** corresponding to each time interval. The components within each eigenbasis are sorted based on the numerical order of their eigenvectors. PCA is then instructed to extract the initial eigenbases that collectively capture a predetermined proportion of the variance in the dataset.

After that, we **quantify the deviation** between eigenbases of two consecutive time intervals. Geometrically, the components selected within each eigenbasis can be conceptualized as new coordinate systems within the reduced observational space. Therefore, it becomes viable to compute the deviation of eigenbases between successive time intervals using cosine similarity.

Cosine similarity is a metric that measures the cosine of the angle between two vectors in a multidimensional space. In the context of comparing eigenbases, cosine similarity quantifies the similarity between the directions represented by the eigenvectors of the respective eigenbases. By calculating the cosine of the angle between these vectors, we obtain a numerical measure of how closely aligned the eigenbases are, indicating the degree of deviation between consecutive time intervals. The cosine similarity $\cos\theta$ between two vectors $\vec{a}$ and $\vec{b}$ is calculated as:

$$\cos\theta \;=\; \frac{\vec{a}\cdot\vec{b}}{||\vec{a}||\cdot||\vec{b}||}$$

As a result of the preceding feature engineering processes, it's possible that certain features may not be considered in every time interval, leading to discrepancies in the feature sets across intervals. This inconsistency poses a challenge when computing cosine similarities. To address this issue, we adopt a strategy where zero-values are introduced into the eigenbases at positions corresponding to the features that are absent in certain intervals. This ensures uniformity in the dimensionality of the eigenbases across all intervals, thus enabling accurate computation of cosine similarities without compromising the integrity of the data.

Given that each eigenbasis consists of multiple vector components, comparing eigenbases of two consecutive time intervals yields several cosine similarities. These cosine similarities represent the degrees of rotation between the axes of distinct but equally dimensional coordinate systems. As these rotations are inherently related, the cosine similarities are unlikely to exhibit significant discrepancies among themselves. To quantify the overall deviation, we first calculate the Euclidean norm of these cosine similarities. Subsequently, we normalize this aggregated norm by dividing it by the square root of the number of vector components. This normalized metric offers a comprehensive indicator of the deviation between consecutive eigenbases, providing a holistic assessment of their alignment in the reduced observational space. Assume vector $\vec{v}$ contains n cosine similarities of two consecutive eigenbases.

$$\vec{v} \;=\; \cos\theta_1, \cos\theta_2, ..., \cos\theta_n$$

Then the deviation indicator Dis based on Euclidean norm is:

$$Dis \;=\; \sqrt{\frac{\cos{\theta_1}^2 + \cos{\theta_2}^2 + ... + \cos{\theta_n}^2}{n}}$$

In the concluding stage of our workflow, we aggregate the deviation values obtained from comparing consecutive eigenbases. These deviation values encapsulate the extent of difference between the equipment's behavior across different time intervals. Subsequently, we employ statistical techniques to fit a probability distribution model to these deviation values. By training this model, we gain insights into the underlying patterns and characteristics of the deviations observed in the equipment's performance.

The obtained probability distribution model serves as a reference for assessing the likelihood of encountering specific deviation magnitudes. Leveraging this model, we derive confidence intervals that provide a range of values within which the true deviation is likely to lie with a certain level of confidence. These confidence intervals offer actionable insights into the uncertainty associated with the deviation estimates, empowering decision-makers to make informed choices regarding maintenance, operational strategies, and resource allocation. Overall, this comprehensive approach enhances our understanding of the equipment's health dynamics and enables proactive measures to mitigate potential risks.

4.4. Approach 2: VARIMA/LSTM + One-class SVM

The VARIMA/LSTM + One-class SVM approach is a powerful method for identifying and detecting anomalous events in various datasets. This technique combines the strengths of the VARIMA/LSTM model, which is capable of capturing time series patterns and trends, with the One-class SVM algorithm, which excels at identifying outliers in high-dimensional spaces.

Moreover, in contrast to the purely statistical PCA-based prognosis method, this approach necessitates a significantly higher computational capacity due to the extensive model training and testing processes involved. Consequently, there arises a requirement to reduce the dimensionality of our datasets, a task that can be efficiently accomplished through Principal Component Analysis.

4.4.1. Principal Component Analysis to Reduce Dimensionality

This step bears resemblance to the PCA-based prognosis method, yet it incorporates an additional step. Following the acquisition of eigenbases based on the feature values of healthy states, which elucidate the significant relationships among the original features, new datasets featuring reduced-dimensional features can be derived from the raw datasets containing data of faulty conditions. Moreover, the outcomes of the eigenbases can be instrumental in classifying the principal factors of the respective system.

4.4.2. Regressors

By capitalizing on the power of regressors, this approach initially forecasts the expected healthy values of features using historical data. These anticipated values act as a benchmark for evaluating the normalcy of incoming data points. In this research, VARIMA and LSTM are employed to execute these tasks, with VARIMA representing a statistical model and LSTM standing as a machine learning approach.

4.4.2.1 VARIMA

At the beginning, we select the first several consecutive time steps of feature data of healthy states for following processes:

Firstly, we initiate the procedure by examining the **stationarity of the input time series data**. In cases where non-stationarity is detected, we apply differencing techniques to stabilize the data, ensuring a solid foundation for subsequent modeling efforts.

Subsequently, we engage in the **model identification** phase, where we ascertain the optimal values for the model's order parameters: p, d, and q. These parameters represent the lag order of the VAR component, the degree of differencing, and the moving average order of the VMA component, respectively.

Following this, we proceed to the **model estimation** stage, employing robust estimation techniques—such as the least squares method or maximum likelihood estimation—to derive the model coefficients accurately.

To ensure the trustworthiness of our model, we carry out meticulous **diagnostic checks** on the model's residuals. These checks evaluate the residuals for normality, homoscedasticity, and the absence of autocorrelation. Should any discrepancies arise, we remain prepared to refine the model as necessary. This study uses the Akaike Information Criterion (AIC) [21] as the basis for the selection of the VARIMA model parameters p and q. In order to determine the best combination of parameters, we first set the range of possible values for p and q. Then, each set of parameter combinations is tested one by one by enumeration and the corresponding AIC value is calculated. In the end, we chose the combination of p and q parameters corresponding to the lowest AIC value as the optimal parameter for the VARIMA model.

With the estimated VARIMA model at our disposal, we can generate informed **forecasts for future time periods**, offering valuable insights into potential trends and patterns that may unfold.

4.4.2.2 LSTM

Initially, we select the initial consecutive time steps of feature data representing healthy states, and process the data with feature engineering for subsequent processing steps:

Firstly, the process begins with further **data preprocessing**. Multi-dimensional time-series data is meticulously divided into three distinct sets: training set, validation set and testing set. This segregation is crucial for evaluating the model's performance without any bias. Normalization techniques are applied to scale the data, ensuring that each feature falls within a comparable range, thus preventing any particular feature from dominating the learning process. The input data is structured so that each input feature has a consistent number of time steps, facilitating a smooth and coherent flow of information through the model.

Secondly, **model definition** comes into play. Utilizing powerful deep learning frameworks (PyTorch in this research), a multi-input multi-output LSTM model is constructed. This model architecture incorporates LSTM layers for each input feature, which may optionally share weights to reduce computational complexity or improve generalization. Fully connected layers or alternative neural network architectures are added subsequent to these layers to refine the LSTM outputs. Multiple output heads are established, each tailored to produce predictions for a specific time-series dimension, catering to the multi-dimensional nature of the predictions.

Thirdly, **parameter initialization** is performed. The model's weight matrices and bias vectors are initialized to initial values, which is critical for the model's convergence during training and can significantly impact its performance.

Afterwards, **forward propagation** takes place, where for each time step in the input sequence, the following operations occur:

1) The current time step data of all input features is passed to their respective LSTM layers.

2) New hidden states are calculated for each LSTM layer based on the previous hidden state and current input.

3) The hidden states of all LSTM layers are passed to subsequent fully connected layers or other network layers.

4) The output is calculated from the fully connected layers or other network layers based on the LSTM hidden states.

5) Predicted values for each dimension are computed using the output heads.

As shown in Figure 4.2, this research contructs a LSTM model with 2 LSTM layers and one fully connected layer. The purpose of using two LSTM layers is to capture complex temporal dependencies in the input time series data. Each LSTM layer processes the input sequence and passes its hidden state to the next layer, allowing the model to learn deeper temporal patterns. The fully connected layer serves as the output layer of the model. It takes the final hidden state from the last LSTM layer as input and maps it to the desired output space. In this case, the fully connected layer is responsible for generating predictions based on the learned temporal information from the LSTM layers.

Figure 4.2: Schematics of LSTM Structure

Then the LSTM model undergoes a series of **iterative processes** to optimize its performance and generate predictions. Initially, loss calculation is performed, with the loss function value calculated based on the predicted outputs and true labels, using common loss functions such as Mean Squared Error (MSE) or Mean Absolute Error (MAE). Subsequently, backpropagation is employed to calculate the gradients of the loss function with respect to the model parameters, utilizing the backpropagation algorithm. This step is vital for optimizing the model's performance.

Following this, **parameter update** occurs, with the model parameters updated using optimization algorithms, such as Gradient Descent or Adam, based on the calculated gradients. This iterative process helps the model learn from its errors and improve over time. The iterative training continues until the desired number of training iterations is reached or a stopping criterion is met, repeating the cycle of forward propagation, loss calculation, backpropagation, and parameter update.

Lastly, model evaluation is conducted by assessing the model's performance on the test set using appropriate evaluation metrics, such as MSE or accuracy rate. This assessment provides insights into the model's generalization capabilities. Now, the trained model is ready for **prediction**, where it leverages its learned patterns and relationships to generate accurate forecasts for new multi-dimensional time-series data.

4.4.3. One-class SVM

The One-class SVM is used to detect any significant deviations between the actual feature values and the predicted healthy values. This technique is particularly effective in identifying anomalies or outliers in the data, as it is designed to model the distribution of normal data points and identify any data points that significantly deviate from this distribution. By comparing the actual feature values to the predicted healthy values, the One-class SVM algorithm helps to identify any potential abnormalities or faults in the system being monitored. This approach enables the early detection of issues, allowing for timely intervention and maintenance, thereby improving the overall reliability and performance of the system.

We initially select a brief sequence of healthy feature data that follows the sequence utilized for regressors. This healthy sequence serves as a valuable reference for training the One-class SVM model, enabling it to learn the typical behavior and patterns exhibited by the system. By employing a distinct sequence for training the One-class SVM, we ensure that the model remains impartial and can accurately detect deviations from the expected healthy state.

Subsequently, we proceed to the Data Preparation phase, where we meticulously clean and normalize both the healthy sequence and the sequence used for regressors. This critical step ensures that the One-class SVM model can effectively learn from the data and make precise predictions. Following cleaning and normalization, we partition the data into training, validation and testing sets, with the training set comprising solely of healthy feature data.

During the Model Training phase, we train the One-class SVM model using the healthy feature data in the training set. Throughout this process, the model learns to discern the boundaries between normal and abnormal data points, empowering it to detect anomalies within the system.

Once the model has been trained, we advance to the Anomaly Detection phase. Here, we apply the trained One-class SVM model to the testing set, which contains a mix of healthy and unhealthy feature data. The model will categorize each data point as normal or abnormal based on its proximity to the decision boundary, aiding us in identifying any potential issues or malfunctions within the system. Then we can further enhance the anomaly detection process by establishing a threshold ratio between abnormal features and all features. This threshold serves as a critical threshold that, when exceeded, triggers an alarm, alerting us to potential issues within the system.

5

Long-term Performance Results

This chapter introduces the results gained from datasets, each of which covers approximately 12 days of second-by-second data to test the performance of all algorithms in long-term time period.

In common sense, those healthy operational conditions should be comprised of stable data. However, we found that in many cases (one example shown in Figure 5.1), there are additional 'abnormal' trips in healthy conditions, consisting of scheduled shutdowns and 'fake' shutdowns caused by other devices. These shutdowns should not be identified by our algorithms. For example, scheduled shutdowns are deliberately designed to help operators make routine inspections and so on, thus should also be included into healthy conditions. But due to the lack of deterministic characteristics to distinguish those extra shutdowns and actual events, our algorithms are likely to mix these two conditions together and trigger unnecessary alarms.

Figure 5.1: One Example of Scheduled Shutdowns (Marked with Green Rectangular Windows) and Actual Event (Marked with Red Rectangular Window)

Here we select 6 events of datasets in total, each of which contains at least one unwanted shutdown. We use these 6 cases to check if our algorithms can distinguish the unwanted shutdowns.

5.1. Results: Water Injection Pump A

Two vibration-related events are found for Water Injection Pump A.

5.1.1. Results: PCA-based Prognosis

As mentioned in Chapter 4.2, the method of PCA-based Prognosis leverages Principal Component Analysis to extract eigenbases, which are then analyzed to detect abnormalities based on deviations from the established correlations observed during normal operational states. The method starts by collecting time series data during the system's healthy state and performing feature engineering, including removing constant features and standardizing data. PCA is then applied to extract the eigenbases of every time interval of datasets (60 time steps per time interval in this research). These eigenbases are used to monitor the system, with deviations from the previous consecutive time interval indicating potential anomalies. Warnings are triggered when these deviations exceed predefined thresholds based on different confidence probabilities and normal deviations in systems' healthy states.

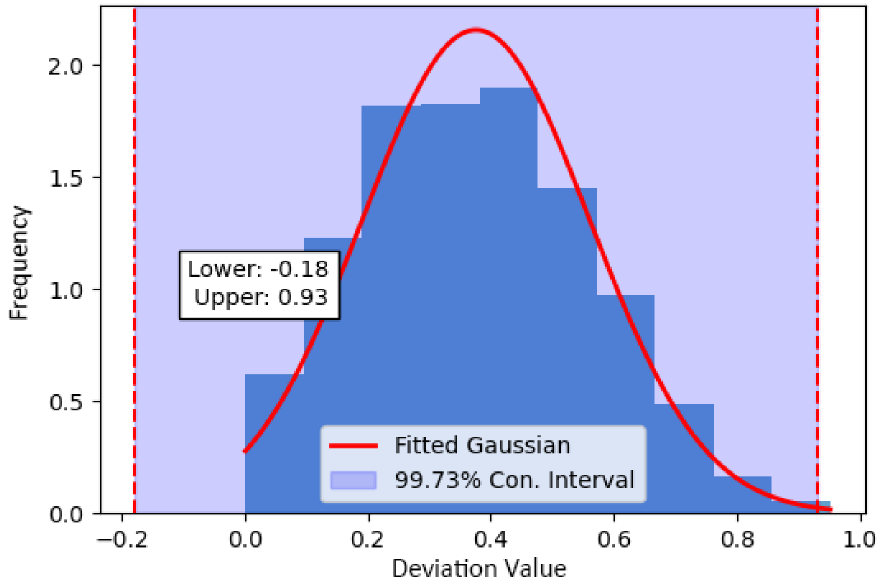

Figure 5.2: Distribution of Eigenbasis Deviations during Normal Conditions CDIEVENT0012021_12DAY

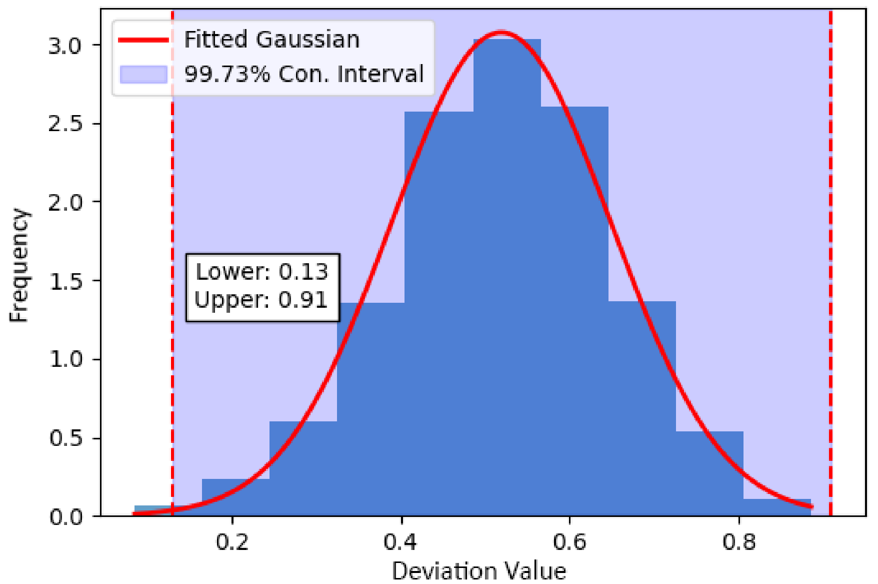

Figure 5.3: Distribution of Eigenbasis Deviations during Normal Conditions CDMEVENT0012022_12DAY

We first collect the eigenbasis deviations when devices are in healthy condition. Above in Figure 5.2 and Figure 5.3 demonstrated the distribution of eigenbasis deviations for the healthy datasets corresponding to the two cases.

As we can see from the two distributions, the layouts of eigenbasis deviations generally follow the Gaussian distribution. And the fitted Gaussian probability distribution functions (Confidence Probability:

99.73%) is also drawn on the figure. The results are basically consistent with the actual situation, because subjectively, when the equipment is running stably, the value of each feature fluctuates around a certain point, and the relationship between features should be the same.

Then we apply 11 different confidence probabilities to the 'healthy deviations', namely [0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 0.9973], and obtain 11 sets of upper and lower limits accordingly.

Then we set these 11 sets of upper and lower limits as the thresholds to trigger false notifications and check the performance of PCA-based Prognosis algorithms via different indicators, including Accuracy and F1 Score. The results are shown in Table 5.1 and Table 5.2.

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.9512	0.5159	0.5713	0.5201	121.7286s
0.91	0.9576	0.5178	0.568	0.5239	119.1522s
0.92	0.9636	0.5203	0.5645	0.5279	116.8872s
0.93	0.9682	0.5226	0.5603	0.5309	124.3213s
0.94	0.9740	0.5291	0.56	0.5382	123.7909s
0.95	0.9782	0.5312	0.5491	0.5378	115.4476s
0.96	0.9814	0.5388	0.5474	0.5426	114.3656s
0.97	0.984	0.5465	0.5422	0.5442	116.2072s
0.98	0.9864	0.5627	0.5402	0.5488	110.6557s
0.99	0.989	0.6089	0.5382	0.5558	110.4962s
0.9973	0.9902	0.6455	0.5193	0.5324	115.2389s

Table 5.1: Performance of PCA-based Prognosis for CDIEVENT0012021_12DAY

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.7948	0.5001	0.5062	0.4467	140.5473s
0.91	0.8106	0.5001	0.5064	0.4516	137.2745s
0.92	0.8251	0.5001	0.5061	0.456	137.0578s
0.93	0.8393	0.4999	0.4978	0.4598	141.7669s
0.94	0.8582	0.5002	0.5073	0.4658	138.1752s
0.95	0.8744	0.5002	0.5078	0.4706	146.2540s
0.96	0.8952	0.5001	0.5029	0.4761	157.9791s
0.97	0.9161	0.4999	0.4981	0.4815	135.4120s
0.98	0.9405	0.4996	0.495	0.4875	144.3665s
0.99	0.9515	0.5045	0.507	0.5048	116.4899s
0.9973	0.9737	0.5098	0.5038	0.5041	123.0880s

Table 5.2: Performance of PCA-based Prognosis for CDMEVENT0012022_12DAY

As we can see from the numerical indicators, the accuracies generally go beyond 95% when the confidence probability reaches up to 99% and 99.73%. However, the F1 scores, which show the comprehensive performance of algorithms regardless of magnitudes, generally vibrate around 0.5. This means that the models may have a trade-off between precision and recall. In other words, among the detection accuracies for healthy time steps and faulty ones, there must be one good and one bad. Two reasons might be that the unrelated shutdowns are mixed within the healthy state, reducing the distinguishing ability of algorithm.

Then we extract the best prediction results from each of the two cases based on the values of F1 scores. As demonstrated in Figure 5.4 and Figure 5.5, a red vertical dotted line divides the entire time period into two parts. The blue part on the left represents that the actual equipment condition is normal, and the blue part on the right represents a fault. Among the detective results shown as the blue line, the time point whose corresponding values are 1 considers the point to be faulty, those with 0 value indicate health.

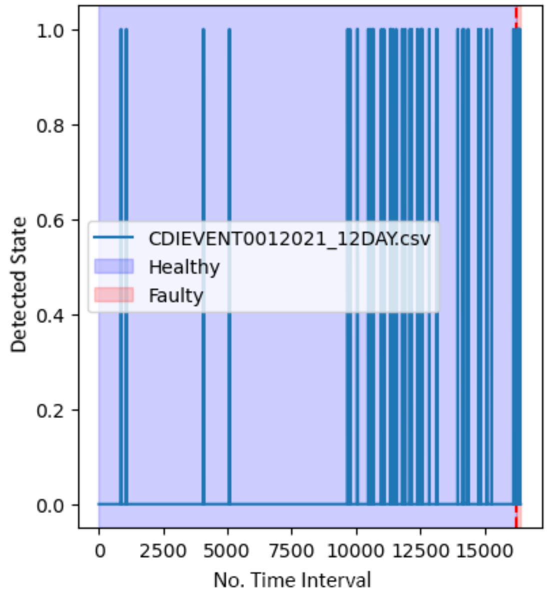

Figure 5.4: CDIEVENT0012021_12DAY: Best Result w.r.t. F1 Score

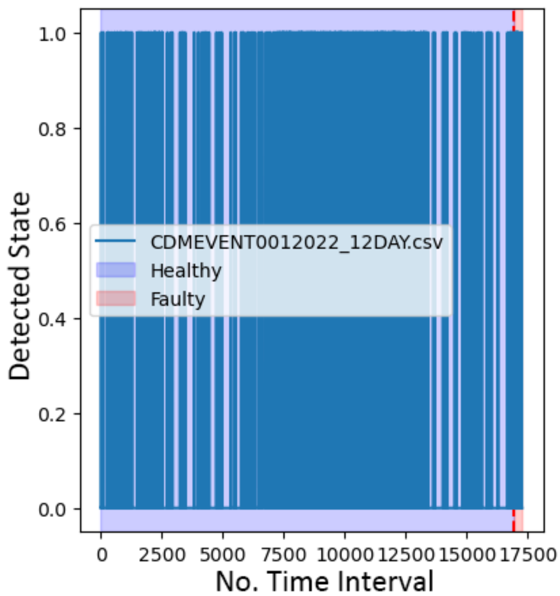

Figure 5.5: CDMEVENT0012022_12DAY: Best Result w.r.t. F1 Score

However, as we can see from the best results, there are still lots of unexpected notifications. Then we check the time intervals of those scheduled shutdowns, and find the majority of them match the time steps where the unexpected notifications are shown. Such coincidence matches our hypothesis. In this way, it can be told that the PCA-based Prognosis is unable to distinguish the scheduled shutdowns from actual trips for both cases.

5.1.2. Results: VARIMA + One-class SVM

The VARIMA + One-Class SVM method begins by applying PCA to reduce the dimensionality of the dataset, which helps improve computational speed; if necessary, the principal components can be analyzed further to interpret what factors they represent. The VARIMA model is then used to capture the dynamics of the system's multivariate time series data, generating predictions for normal operating conditions. The residuals, calculated as the difference between the predicted and actual values, reflect any deviations from expected behavior. Features are extracted from these residuals and used to train a One-Class SVM model with data from normal operating conditions. During deployment, new residual features are fed into the trained One-Class SVM, which identifies any deviations from the normal range, effectively detecting and signaling anomalies in the system.

Since PCA is used solely to reduce the dimensionality of the datasets and interpreting the principal components is not essential for this approach, we present the principal component information for these two cases as examples, as shown in Table 5.3 and Table 5.4. Generally, to interpret the meaning of each principal component, we look at the features with the highest positive and highest negative loadings in the factor analysis (Loadings whose values difference are within 0.01 compared with the highest or lowest ones are also considered, since their contributions are highly similar). A positive loading indicates that the feature is positively correlated with the principal component, while a negative loading indicates a negative correlation. By analyzing these loadings, we can better understand the combination of variables each principal component represents and their role in the data. Ideally, once one alarm is triggered, operators can refer to the exact principal component that first show abnormality and seek for the source of problems accordingly.

The principal components in the above two examples reveal that the operational conditions differ between FPSOs, even though the system structures are identical. Additionally, based on previous surveys, the operational patterns of a single system can also change over time.

	Main Features		Description	Factor
1st Principal Component	Most Positive	T26-VXI-1820	Motor DE Radial Bearing Vibration 'x'	0.54
	Most Negative	T26-VYI-1828	Motor NDE Radial Bearing Vibration 'y'	-0.068
2nd Principal Component	Most Positive	T26-VYI-1821	Motor DE Radial Bearing Vibration 'y'	0.39
	Most Negative	T26-VXI-1827	Motor NDE Radial Bearing Vibration 'x'	-0.51
3rd Principal Component	Most Positive	T26-VXI-1811	Pump NDE Radial Bearing vibration 'x'	0.75
	Most Negative	T26-VYI-1812	Pump NDE Radial Bearing vibration 'y'	-0.5
4th Principal Component	Most Positive	T26-VXI-1820	Motor DE Radial Bearing Vibration 'x'	0.7
	Most Negative	T26-VYI-1812	Pump NDE Radial Bearing vibration 'y'	-0.57
5th Principal Component	Most Positive	T26-VYI-1821	Motor DE Radial Bearing Vibration 'y'	0.64
	Most Negative	T26-VYI-1812	Pump NDE Radial Bearing vibration 'y'	-0.4

Table 5.3: Principal Components for CDIEVENT0012021_12DAY (Variance Ration:95%)

	Main Features		Description	Factor
1st Principal Component	Most Positive	T26-VYI-1828	Motor NDE Radial Bearing Vibration 'y'	0.47
	Most Negative	T26-VYI-1812	Pump NDE Radial Bearing vibration 'y'	-0.33
2nd Principal Component	Most Positive	T26-VXI-1815	Pump DE Radial Bearing vibration 'x'	0.33
	Most Negative	T26-VXI-1827	Motor NDE Radial Bearing Vibration 'x'	-0.53
3rd Principal Component	Most Positive	T26-VXI-1815	Pump DE Radial Bearing vibration 'x'	0.55
	Most Negative	T26-VYI-1816	Pump DE Radial Bearing vibration 'y'	-0.41
4th Principal Component	Most Positive	T26-VXI-1815	Pump DE Radial Bearing vibration 'x'	0.6
		T26-VYI-1816	Pump DE Radial Bearing vibration 'y'	0.6
	Most Negative	T26-VYI-1821	Motor DE Radial Bearing Vibration 'y'	-0.32
5th Principal Component	Most Positive	T26-VYI-1816	Pump DE Radial Bearing vibration 'y'	0.56
	Most Negative	T26-VXI-1827	Motor NDE Radial Bearing Vibration 'x'	-0.59

Table 5.4: Principal Components for CDMEVENT0012022_12DAY (Variance Ration:95%)

Since the amounts of features considered in this research are no higher than 10 for each device, we put more focus on remaining such much information as possible than reducing the numbers of original features. Otherwise, the considerable loss of information may highly influence the detective results. For all the 6 cases, we ask our algorithm to extract the first several principal components whose accumulative variance ratio are no larger than 95%. Then in both cases of this section, the dimensionality of the original 8 features are reduced to 5 new features.

Since all the values of features at the start of testing sequences considered healthy are vibrating stably and demonstrate highly similar patterns, thus we do not need long sequences of samples to train the VARIMA model. However, in order to help computer automatically obtain such generic thresholds as possible, we need to feed One-class SVM model with such large amounts of samples as possible. Following the idea, we then use the first 1/120 of each healthy sequence to train the VARIMA models and the next 1/20 to train the One-class SVM models (Ratios can change with respect to the actual need of operators). The performance indicators are shown in Table 5.5 and Table 5.6. The VARIMA and One-class SVM models give different results after training. The former gives the result at once, while the latter can give the judgment result at each time step. In other words, the model training time and the time required to give the detection result of this method should be considered separately. Therefore, we use two additional indicators to measure the time cost of this method, namely *Model Training* and *Detection per Time Step*. The same is true for the approach of LSTM + One-class SVM in the following chapters. In both cases, the accuracy decreases compared to PCA-based Prognosis. For the first case, the F1 score increases, indicating better overall performance than PCA-based Prognosis. However, the second case shows the opposite trend.

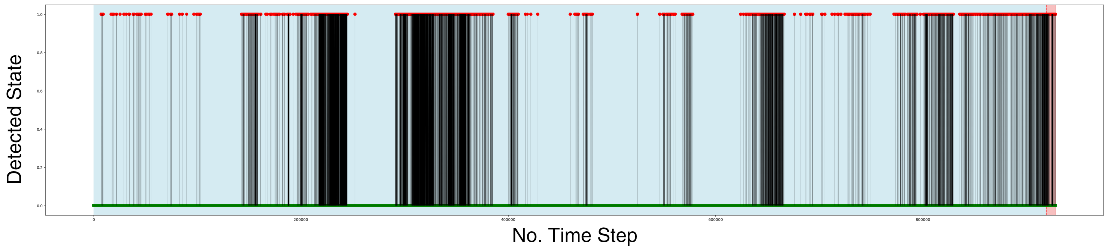

Figure 5.6: VARIMA + One-class SVM: Detection results of CDIEVENT0012021_12DAY

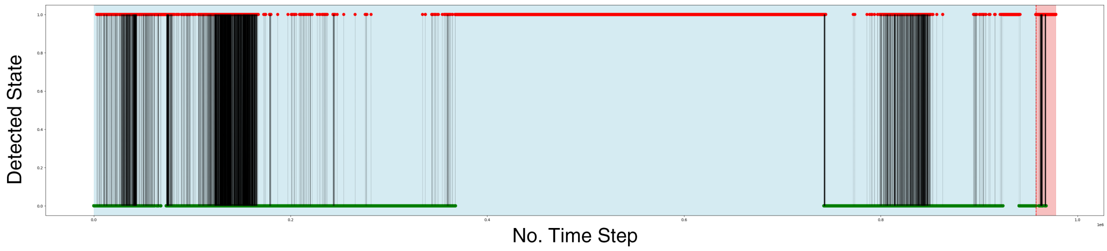

Figure 5.7: VARIMA + One-class SVM: Detection results of CDMEVENT0012022_12DAY

The detection results are shown in Figure 5.6 and Figure 5.7 for better and more direct understanding of performance. And now it becomes quite clear that lots of unexpected notifications are shown before the actual trip, and the number of those 'false alarms' for the second case significantly excels that of the first case. We then examine the distribution of the extra and unwanted shutdowns in each case to understand the performance differences. In the first case, those shutdowns account for around 2 days of data (1/6 of the dataset), whereas in the second case, they extend to around 6 days (half of the dataset). This increase in noise not only results in more unexpected notifications but also affects the model parameters. The VARIMA + One-class SVM approach is sensitive to abnormal shifts in data values, making it more susceptible to scheduled shutdowns compared to PCA-based Prognosis. This sensitivity primarily explains the differing performance of our algorithms in the two cases.

5.2. Results: Water Injection Pump B

Two vibration-related events are found for Water Injection Pump B.

5.2.1. Results: PCA-based Prognosis

Following the same steps mentioned in section 5.1, the collection and analysis on eigenbasis deviations in healthy states are first conducted. As shown in Figure 5.8 and Figure 5.9, the distributions of

Accuracy	Precision	Recall	F1
0.9436	0.5465	0.7629	0.5688
Model Training/s		Abnormality Detection/s	
87.8101		1509.5243	
Detection Steps		Dectection per Time Step	
976358		0.001546077	

Table 5.5: Performance of VARIMA + One_class SVM for CDIEVENT0012021_12DAY

Accuracy	Precision	Recall	F1
0.5767	0.5185	0.726	0.4022
Model Training/s		Abnormality Detection/s	
98.5687		1662.3902	
Detection Steps		Dectection per Time Step	
1028329		0.001616594	

Table 5.6: Performance of VARIMA + One_class SVM for CDMEVENT0012022_12DAY

correlation deviations of healthy data for the two cases also follow Gaussian distributions.

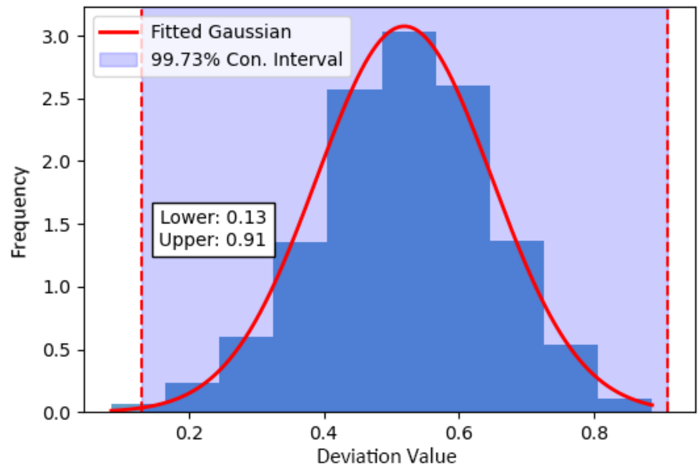

Figure 5.8: Distribution of Eigenbasis Deviations during Normal Conditions CDSEVENT0052022_12DAY

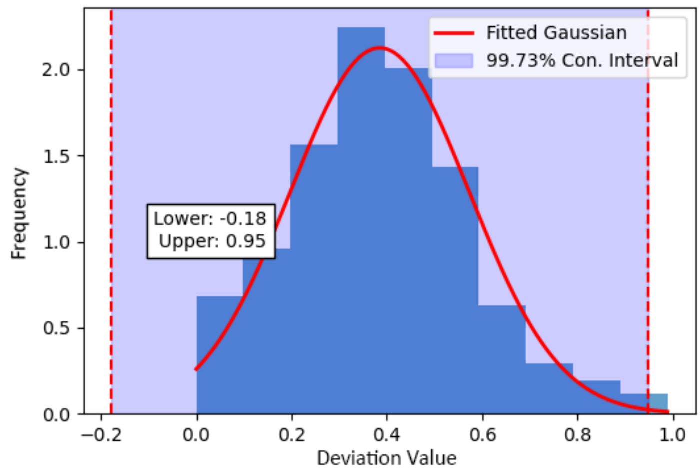

Figure 5.9: Distribution of Eigenbasis Deviations during Normal Conditions CDMEVENT0062023_12DAY

Then still, the same 11 confidence probabilities are picked to get upper/lower limits as thresholds according to the collected values and test the performance of PCA-based Prognosis algorithm. The results are shown in table 5.7 and 5.8. Upon examining the results, it becomes evident that the performance of the algorithm for the two cases of Water Injection Pump B is generally comparable to that for the cases of Water Injection Pump A. Although the accuracy surpasses 95% when the confidence probability exceeds 98%, the F1 scores fluctuate around 0.5, which suggests a suboptimal ability to discriminate between healthy and faulty conditions.

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.8643	0.4939	0.4837	0.4823	26.6844s
0.91	0.8746	0.4928	0.4825	0.4831	26.5637s
0.92	0.9023	0.4936	0.489	0.4898	26.7154s
0.93	0.9023	0.4936	0.489	0.4898	26.7154s
0.94	0.9136	0.4918	0.4883	0.4896	26.5672s
0.95	0.925	0.4888	0.4877	0.4882	26.3715s
0.96	0.9357	0.49	0.4919	0.4909	26.4581s
0.97	0.9422	0.4926	0.4953	0.4934	26.4184s
0.98	0.9491	0.4945	0.4976	0.4945	26.7565s
0.99	0.9564	0.5004	0.5	0.4955	26.4406s
0.9973	0.9618	0.4818	0.4989	nan	26.549s

Table 5.7: Performance of PCA-based Prognosis for CDSEVENT0052022_12DAY

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.8849	0.4933	0.4837	0.4845	48.8452s
0.91	0.8928	0.4933	0.4853	0.4863	48.48s
0.92	0.9013	0.4939	0.488	0.4888	61.5822s
0.93	0.9096	0.4941	0.4898	0.4906	48.9816s
0.94	0.9170	0.4942	0.4911	0.4919	49.4838s
0.95	0.9262	0.4933	0.4916	0.4923	48.6251s
0.96	0.9353	0.4954	0.4954	0.4954	48.9537s
0.97	0.94	0.4966	0.4971	0.4967	48.9184s
0.98	0.9464	0.4941	0.4961	0.4947	48.7635s
0.99	0.9536	0.4978	0.499	0.4967	48.6766s
0.9973	0.9631	0.5177	0.5023	0.4983	49.1909s

Table 5.8: Performance of PCA-based Prognosis for CDMEVENT0062023_12DAY

As shown in the figures of best results (Figure 5.10 and Figure 5.11) according to F1 scores, there are numerous premature notifications and the algorithm fails to detect the actual start of the trips of both two cases. This strategy that employs cosine similarity of eigenbase differences for correlation deviation identification still struggles to accurately capture the abnormal variations in feature correlations for these two cases due to ambiguous abnomral changes in eigenbasis deviations and the similar reasons of the previous two cases.

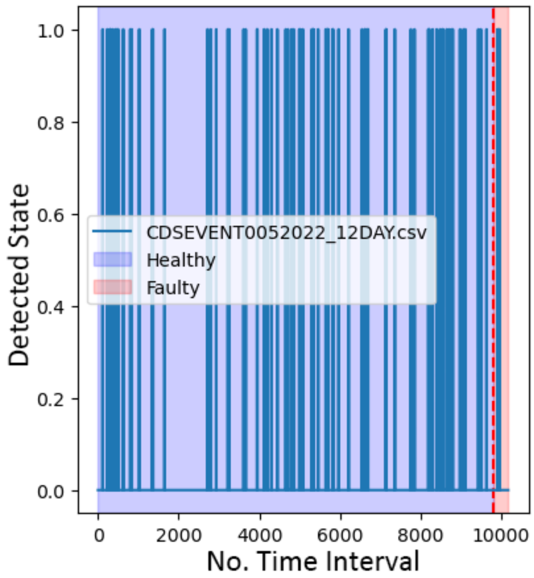

Figure 5.10: CDSEVENT0052022_12DAY: Best Result w.r.t. F1 Score

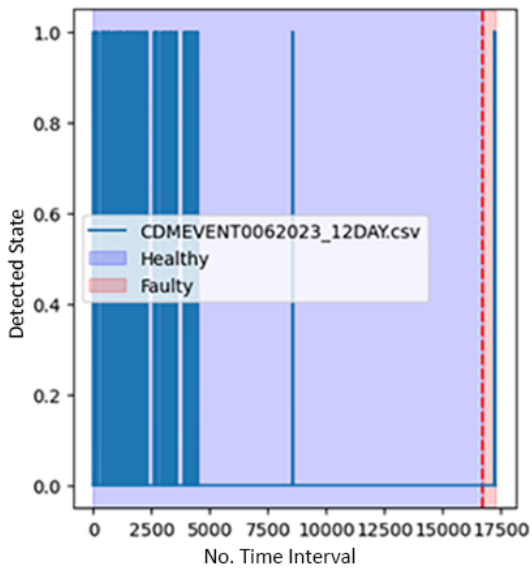

Figure 5.11: CDMEVENT0062023_12DAY: Best Result w.r.t. F1 Score

5.2.2. Results: VARIMA + One-class SVM

Same as the steps mentioned in section 5.1, in both cases, we apply PCA to reduce the dimensionality of the original 8 features to 4 new features, capturing 95% of the variance. We then use the first 1/120 of each healthy sequence to train the VARIMA models and the following 1/20 of each healthy sequence to train the One-class SVM models. The performance indicators are shown in Table 5.9 and Table 5.10. And the detection results are shown in Figure 5.6 and Figure 5.7.

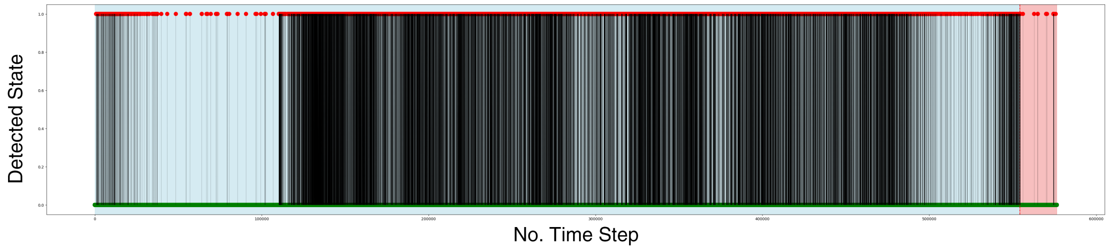

Figure 5.12: VARIMA + One-class SVM: Detection results of CDSEVENT0052022_12DAY

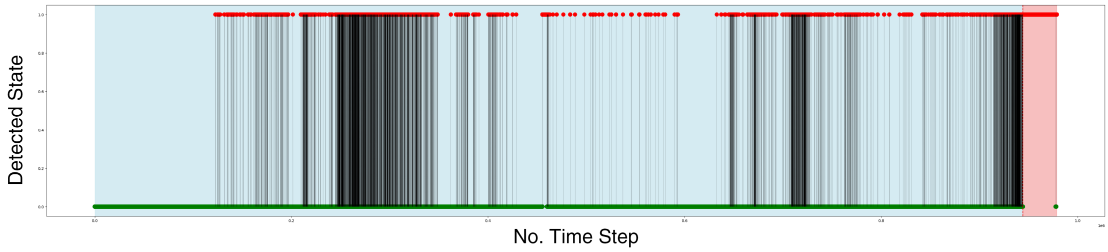

Figure 5.13: VARIMA + One-class SVM: Detection results of CDMEVENT0062023_12DAY

Accuracy	Precision	Recall	F1
0.9363	0.4816	0.4877	0.4846
Model Training/s		Abnormality Detection/s	
441.1361		824.4326	
Detection Steps		Detection per Time Step	
605542		0.001361479	

Table 5.9: Performance of VARIMA + One_class SVM for CDSEVENT0052022_12DAY

Accuracy	Precision	Recall	F1
0.9856	0.8556	0.9923	0.9117
Model Training/s		Abnormality Detection/s	
609.5172		1951.6148	
Detection Steps		Detection per Time Step	
1028448		0.001897631	

Table 5.10: Performance of VARIMA + One_class SVM for CDMEVENT0062023_12DAY

For the first case, the accuracy exceeds 90%, but the F1 score remains below 0.5. In contrast, the F1 score for the second case surpasses 90%, indicating a better overall performance compared to PCA-based Prognosis. We then examine the distribution of extraneous, unrelated shutdowns in each case to explain the performance differences. In the first case, shutdowns unrelated to vibration issues constitute around 1 day of data (approximately 1/12 of the dataset), whereas in the second case, these shutdowns decrease to about 2 hours (less than 0.7% of the dataset). This reduction in noise not only results in fewer unexpected notifications but also influences the model parameters. Besides, the data of those unrelated shutdowns also enters the model-training dataset for the first case, which largely affects the accuracy and distinguishing ability of the model. The VARIMA + One-class SVM approach is sensitive to abnormal shifts in data values, making it more susceptible to unrelated shutdowns compared to PCA-based Prognosis. This sensitivity largely accounts for the differing performance of our algorithms in the two cases.

5.3. Results: HP Feed Pump A/B

Given that there is only one single case for HP Feed Pump A and B respectively, this chapter presents the findings from both cases in a consolidated manner, namely CDSEVENT0072022_12DAY and CDSEVENT0102023_12DAY.

5.3.1. Results: PCA-based Prognosis

As shown in Figure 5.14 and Figure 5.15, the distributions of correlation deviations for healthy data in both cases follow Gaussian distributions. We used the same 11 confidence probabilities to test the performance of the PCA-based Prognosis algorithm. The results are presented in Table 5.11 and Table 5.12.

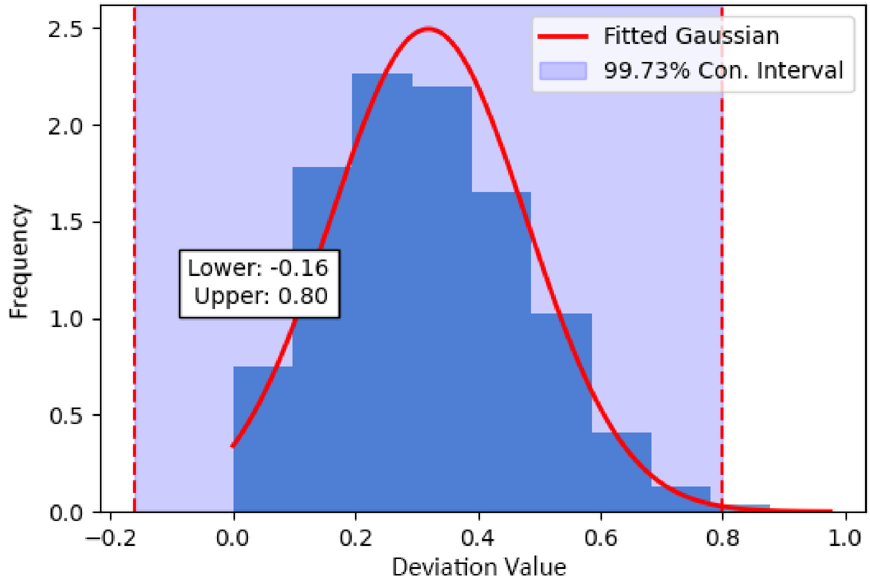

Figure 5.14: Distribution of Eigenbasis Deviations during Normal Conditions CDSEVENT0072022_12DAY

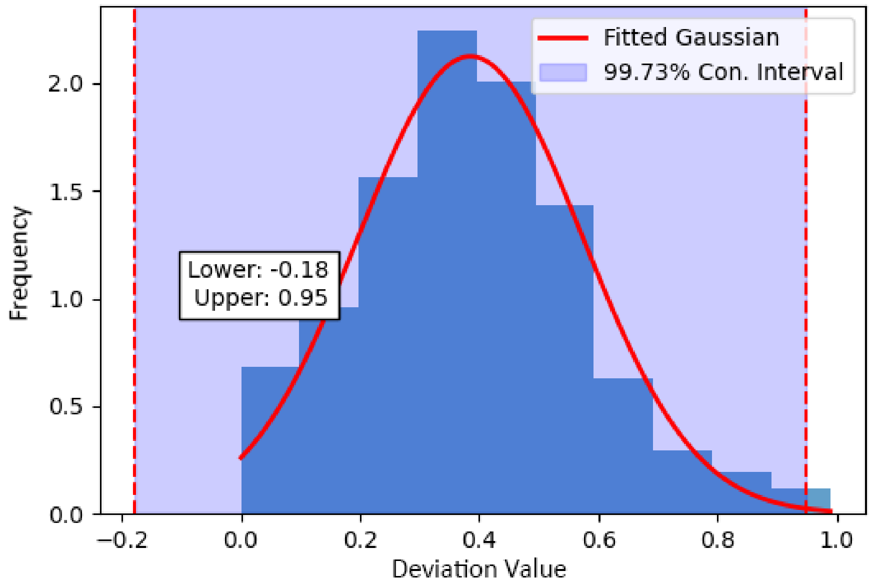

Figure 5.15: Distribution of Eigenbasis Deviations during Normal Conditions CDSEVENT0102023_12DAY

Examining the results reveals that the algorithm's performance for the two cases of HP Feed Pump

A/B is generally comparable to that of the previous two subsystems. Although the accuracy exceeds 95% when the confidence probability is above 98%, the F1 scores fluctuate around 0.5, indicating a suboptimal ability to differentiate between healthy and faulty conditions.

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.952	0.5055	0.5319	0.5029	52.4134s
0.91	0.9577	0.5046	0.5227	0.5026	52.1794s
0.92	0.9631	0.5051	0.5213	0.5044	52.064s
0.93	0.968	0.5067	0.5238	0.5077	51.9475s
0.94	0.9726	0.5062	0.518	0.5075	51.8754s
0.95	0.9773	0.5072	0.5163	0.5091	51.789s
0.96	0.9805	0.5077	0.5139	0.5095	59.1238s
0.97	0.9838	0.5117	0.5156	0.5133	52.4985s
0.98	0.9868	0.5101	0.509	0.5095	52.2126s
0.99	0.9897	0.5137	0.5064	0.5084	53.7749s
0.9973	0.9918	0.544	0.5075	0.5118	53.6125s

Table 5.11: Performance of PCA-based Prognosis for CDSEVENT0072022_12DAY

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.9034	0.5065	0.5557	0.4941	47.9865s
0.91	0.914	0.5072	0.5552	0.4981	47.7104s
0.92	0.9252	0.5074	0.5492	0.5013	47.8197s
0.93	0.9364	0.5067	0.5372	0.5028	48.6632s
0.94	0.9467	0.5074	0.5337	0.5061	48.8119s
0.95	0.9567	0.5101	0.5358	0.5119	48.223s
0.96	0.9634	0.5126	0.5362	0.5162	49.92s
0.97	0.9688	0.5141	0.5331	0.5184	47.1706s
0.98	0.9745	0.5141	0.5243	0.5174	48.1269s
0.99	0.9811	0.5157	0.5159	0.5158	47.3453s
0.9973	0.9864	0.516	0.5068	0.509	47.218s

Table 5.12: Performance of PCA-based Prognosis for CDSEVENT0102023_12DAY

Figure 5.16 and Figure 5.17 show the best results according to F1 scores. However, these figures also highlight numerous premature notifications and the algorithm's failure to detect the actual onset of the trips in both cases. Therefore, in addition to the impact of irrelevant trips, the strategy that uses cosine similarity of eigenbase differences for correlation deviation identification struggles to accurately capture variations in feature correlations.

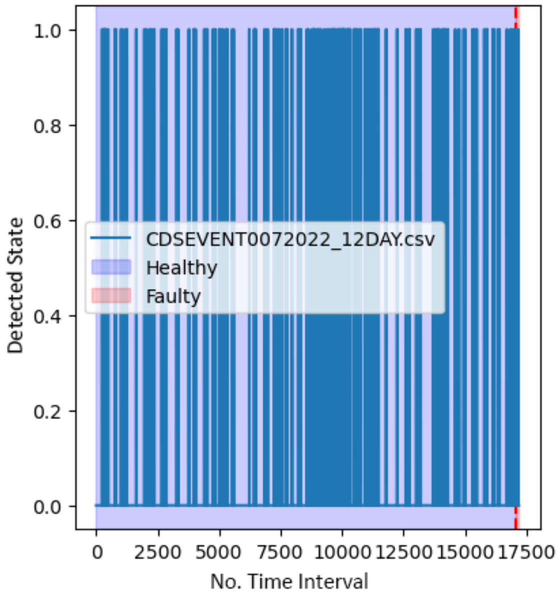

Figure 5.16: CDSEVENT0072022_12DAY: Best Result w.r.t. F1 Score

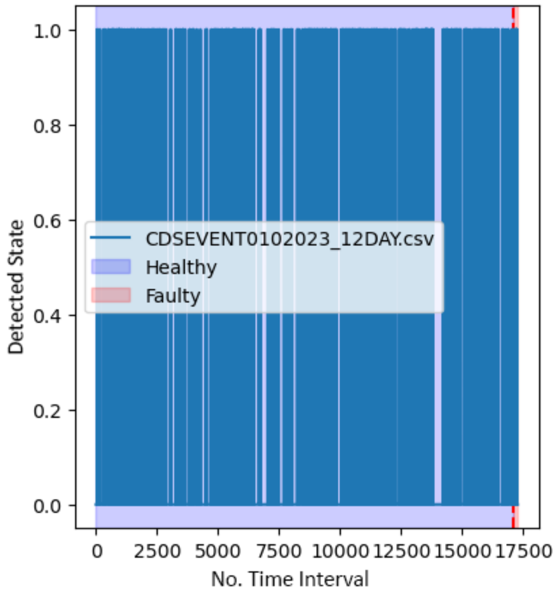

Figure 5.17: CDSEVENT0102023_12DAY: Best Result w.r.t. F1 Score

5.3.2. Results: VARIMA + One-class SVM

In both cases, we apply PCA to reduce the dimensionality of the original features to 7 new features for both two devices, capturing 95% of the variance. We then use the first 1/120 of each healthy sequence to train the VARIMA models and the following 1/20 of each healthy sequence to train the One-class SVM models. The performance indicators are shown in Table 5.13 and Table 5.14. And the detection results are shown in Figure 5.18 and Figure 5.19.

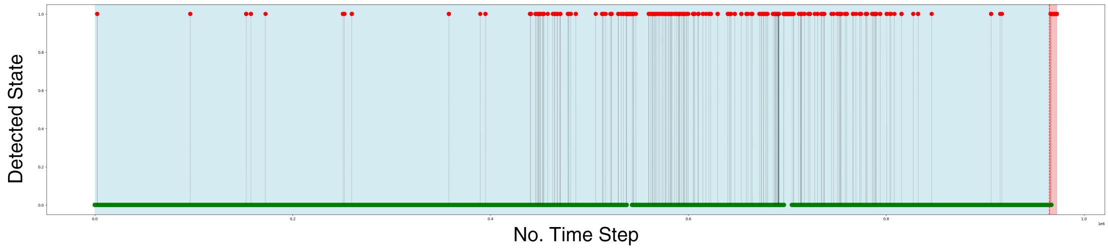

Figure 5.18: VARIMA + One-class SVM: Detection results of CDSEVENT0072022_12DAY

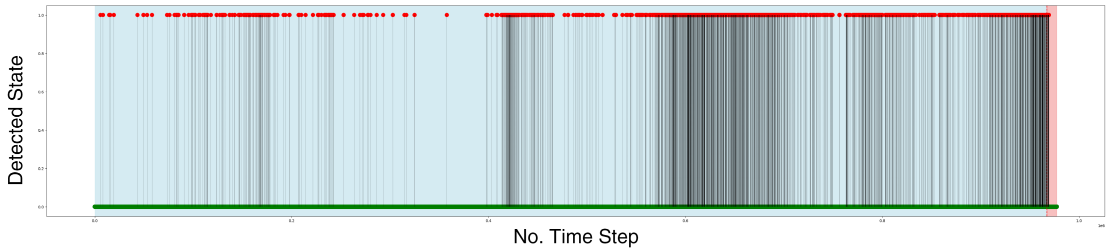

Figure 5.19: VARIMA + One-class SVM: Detection results of CDSEVENT0102023_12DAY

Accuracy	Precision	Recall	F1
0.9832	0.6366	0.85974	0.696
Model Training/s		Abnormality Detection/s	
907.3419		2655.9532	
Detection Steps		Dectection per Time Step	
1023309		0.002595456	

Table 5.13: Performance of VARIMA + One_class SVM for CDSEVENT0072022_12DAY

Accuracy	Precision	Recall	F1
0.9867	0.502	0.5005	0.4998
Model Training/s		Abnormality Detection/s	
708.9208		2379.4813	
Detection Steps		Dectection per Time Step	
1028243		0.002314124	

Table 5.14: Performance of VARIMA + One_class SVM for CDSEVENT0102023_12DAY

In the second case, the accuracy is above 90%, but the F1 score remains below 0.5. Conversely, the first case has an F1 score of approximately 70%, indicating better overall performance.

To explain the performance differences, we examined the distribution of extraneous, unrelated shutdowns in each case. In the first case, unrelated shutdowns accounted for around 7 hours of data (approximately 2% of the dataset), whereas in the second case, these shutdowns extended to about 1 day (around 1/6 of the dataset). This increase in noise not only led to fewer unexpected notifications but also influenced the model parameters. The VARIMA + One-class SVM approach is more sensitive to abnormal data shifts compared to PCA-based Prognosis. This sensitivity largely explains the differing performance of our algorithms in the two cases.

The heightened sensitivity to unrelated shutdowns in the VARIMA + One-class SVM approach underscores the importance of noise reduction for accurate anomaly detection. Minimizing the impact of irrelevant shutdowns is crucial for enhancing model robustness and ensuring reliable fault detection.

In general, while the performance of VARIMA+One-class SVM is better than that of PCA-based Prognosis, both methods are easily affected by irrelevant shutdowns. Additionally, for PCA-based Prognosis, some results indicate that using the deviation of eigenbasis based on Euclidean distance as the indicator may not clearly reflect abnormal changes when the real vibration-related trips occur.

6
Short-term Performance Results

As discussed in Chapter 5, those unwanted shutdowns ahead of the actual trips have the potential to introduce unnecessary additional notifications, which can significantly compromise the precision of our algorithms. Furthermore, the shift in operational patterns preceding and following these shutdowns can also disrupt the outcomes. To mitigate the impact of these distracting elements, our datasets must be partitioned based on failure time boundaries and scrutinized autonomously for segments that exclude scheduled shutdowns. This approach effectively emulates the actions of operators who consciously disregard the shutdowns not related to vibration and reinitialize the algorithms subsequent to each trip. By doing so, our code parameters can adapt to the evolving state of the system, thereby enhancing the reliability of the monitoring results. This chapter introduces the results of those datasets including scheduled shutdowns using such strategy.

Besides, there are other factors can also significantly influence the results, such as the twist in original data characteristics caused by feature engineering and the false warnings caused by the issue that the additionally selected health data is not representative of the test set's health status criteria. Thus, further modifications of algorithms are made, which are listed below:

1. Removed the irrelevant shutdowns;

2. (PCA-based prognosis) Used the first time interval of healthy periods as the baseline of eigenbasis deviation;

3. (PCA-based prognosis) Removed cosine similarity of eigenbasis and replaced it by the Euclidean distance of similarity matrix/correlation matrix (Refering to the work of Elmore & Richman [22]);

4. Selected the first 1/40 of the 'healthy' as the training sets to give thresholds of confidence intervals.

5. (PCA-based prognosis) Only considered upper limits, since low values of deviations mean the status of device is close to the original healthy baseline even though exceling the lower thresholds.

6. (VARIMA/LSTM + One-class SVM) removed PCA from the procedure of feature engineering, since there are only 6-8 vibration-related features for each device.

We select the same but shorter-term cases for each device, and check the changes in performance with this new approach. Besides, the decreases in numbers of time steps make it possible to deploy LSTM as one of the regressors to extract the healthy characteristics of data, since it usually take long time to train recurrent neural network models with large-scale data, making it unfeasible to be deployed in real industrial environments.

6.1. Results: PCA-based Prognosis

After adjustments, the PCA-based Prognosis method begins by gathering time series data from the system's healthy state and conducting feature engineering, which includes removing constant features and standardizing the data. PCA is then employed to extract the correlation matrix for each time interval in the datasets (60 time steps per interval in this study). These correlation matrices are utilized for system monitoring, with deviations from the initial time interval in the healthy state signaling potential anomalies. Warnings are issued when these deviations surpass predefined thresholds, which are determined based on various confidence probabilities and normal deviations observed in the system's healthy states.

Demonstrated in Table 6.1 and Table 6.2 are the performance of PCA-based Prognosis for case CDMEVENT0012022_12DAY and case CDMEVENT0012022 when confidence probability is 99% (Confidence probability corresponding to the highest F1 score in long-term performance). It is clear that the accuracy of the new case decreases while the performance of distinguishing (indicated as F1 score) increases from 0.5 to over 0.6.

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.99	0.9515	0.5045	0.507	0.5048	116.4899s

Table 6.1: Performance of PCA-based Prognosis for CDMEVENT0012022_12DAY

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.99	0.9193	0.7739	0.5829	0.6158	10.4746s

Table 6.2: Performance of PCA-based Prognosis for CDMEVENT0012022

Then we check the results of detection for the new case shown in Figure 6.1, and it is obvious that the unexpected notifications are less dense than that of long-term performance. The reduction of affects by irrelevant shutdowns is the main reason of such changes.

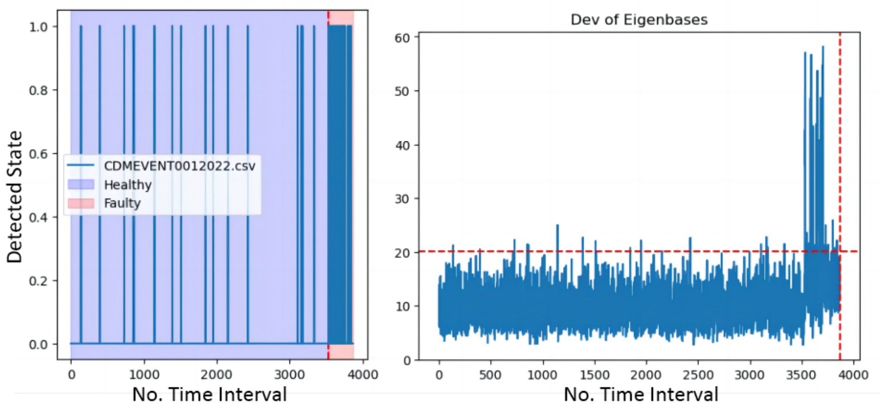

Figure 6.1: CDMEVENT0012022: Results when Confidence Prob. = 99%

Furthermore, we check the performance of the new case for all confidence probabilities (table 6.3), and now it can be told that the comprehensive performance of PCA-based Prognosis generally increases compared with that of the 12-day case. It indirectly proves the large extent of influence of unwanted shutdowns within the long-term datasets.

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.894	0.6663	0.669	0.6677	10.3817s
0.91	0.8958	0.6674	0.6606	0.6639	10.7895s
0.92	0.9007	0.6792	0.6606	0.6691	10.4936s
0.93	0.9056	0.692	0.6565	0.6717	10.8699s
0.94	0.9080	0.6989	0.6537	0.6723	10.3705s
0.95	0.9113	0.7099	0.6447	0.6693	10.4926s
0.96	0.9136	0.7196	0.6406	0.6686	10.3302s
0.97	0.9147	0.7239	0.6223	0.6536	10.1927s
0.98	0.9175	0.7444	0.6062	0.6408	10.2875s
0.99	0.9193	0.7739	0.5829	0.6158	10.4746s
0.9973	0.9188	0.7974	0.5583	0.5821	10.3982s

Table 6.3: Short-term Performance of PCA-based Prognosis for CDMEVENT0012022

Similar results are observed in case CDMEVENT0062023, as shown by the performance indicators in Table 6.4. Similar to the previous case, the F1 scores increase to around 0.6, while the accuracies generally decrease slightly. The reason of such increases in performance generally lies in the evident abnormal changes when and before the time interval where actual trip happens, which matches our principle.

Confidence Prob.	Accuracy	Precision	Recall	F1	Running Time
0.9	0.9325	0.5715	0.7099	0.5995	19.909s
0.91	0.9363	0.5748	0.7063	0.6036	19.9509s
0.92	0.9383	0.576	0.7017	0.6047	19.6592s
0.93	0.9408	0.5756	0.6891	0.6032	19.8984s
0.94	0.9435	0.5748	0.6738	0.6007	19.9055s
0.95	0.9459	0.5744	0.6612	0.5986	20.0527s
0.96	0.9485	0.5741	0.6486	0.5965	19.9947s
0.97	0.9518	0.5744	0.6336	0.594	22.0865s
0.98	0.9552	0.5738	0.6159	0.5893	22.052s
0.99	0.9619	0.5833	0.5999	0.5907	24.616s
0.9973	0.968	0.6048	0.5919	0.5979	21.9329s

Table 6.4: Short-term Performance of PCA-based Prognosis for CDMEVENT0062023

A more detailed inspection of the detection results shown in Figure 6.2 reveals that dense false notifications are triggered not only when/after but also several minutes before the actual trip. It means that the abnormal shifts within the correlations of features becomes clear right before the actual trip, which are captured by the algorithm. Although in the two cases mentioned above, the indicators

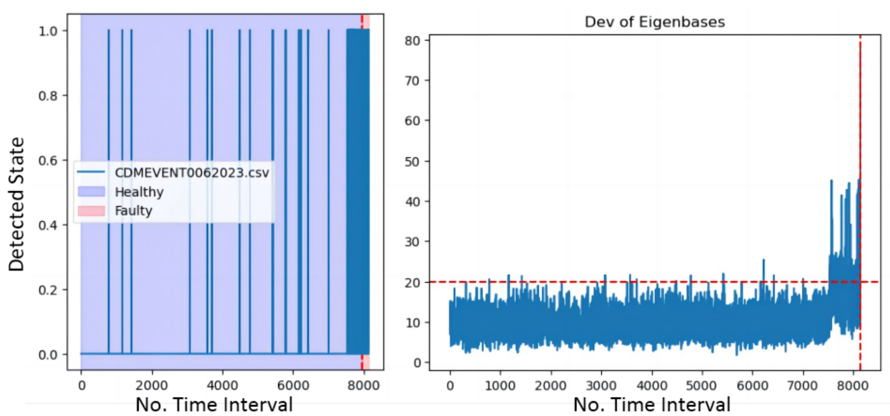

Figure 6.2: CDMEVENT0062023: Results when Confidence Prob. = 99%

showing changes in the relationships between system components exhibited clear abnormal fluctuations at or before the actual moment of system trip, such abnormal fluctuations were still not obvious in most other cases. Demonstrated below in Figure 6.3, 6.4, 6.5 and 6.6 are the detection results for case CDIEVENT0012021, CDSEVENT0052022, CDSEVENT0072022 and CDSEVENT0102023 respectively.

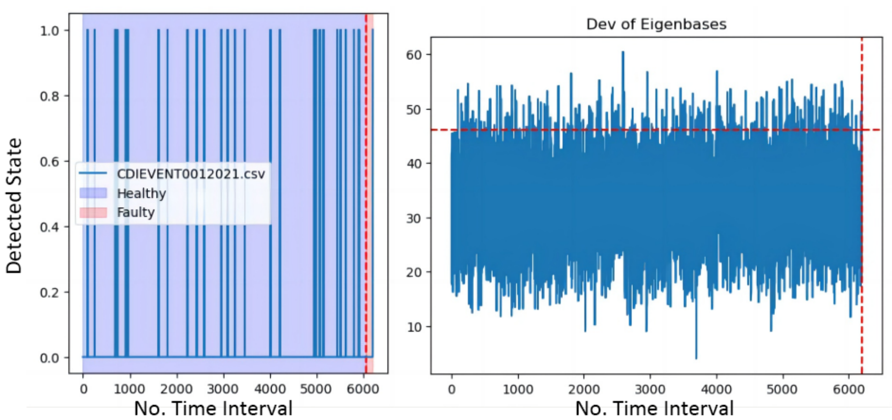

Figure 6.3: CDIEVENT0012021: Results when Confidence Prob. = 99%

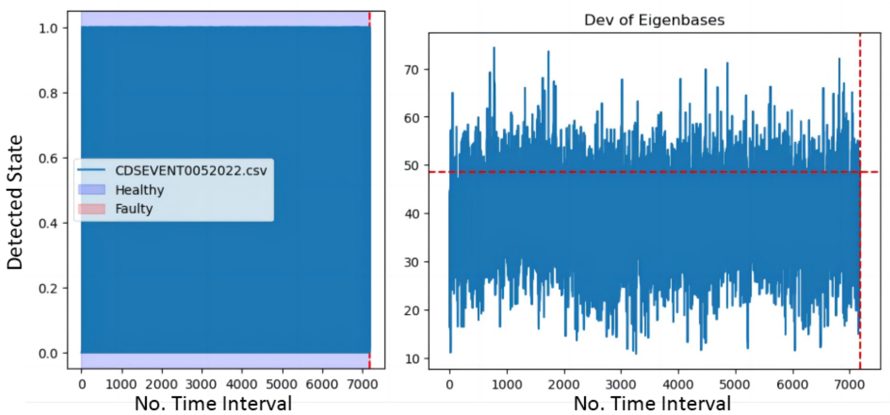

Figure 6.4: CDSEVENT0052022: Results when Confidence Prob. = 99%

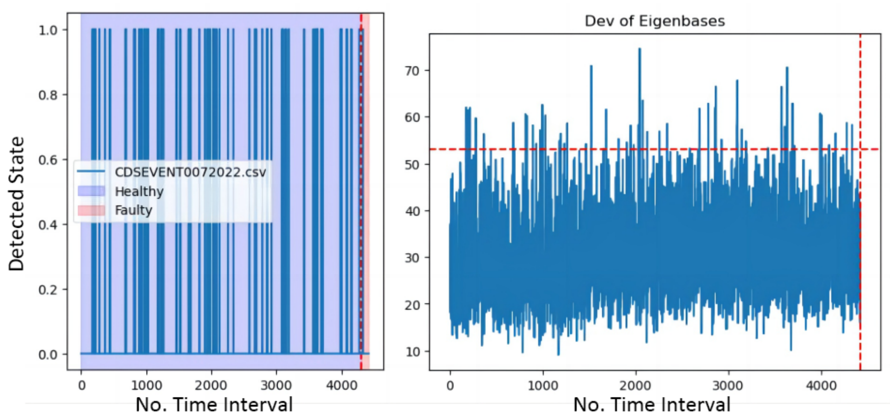

Figure 6.5: CDSEVENT0072022: Results when Confidence Prob. = 99%

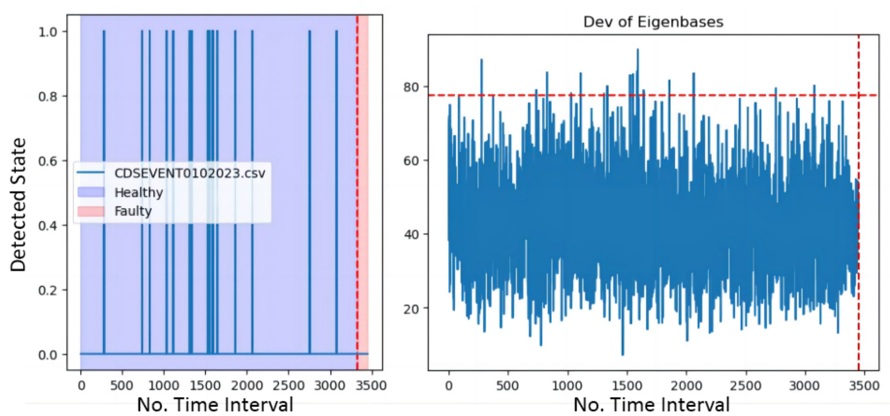

Figure 6.6: CDSEVENT0102023: Results when Confidence Prob. = 99%

As seen from these detection results, abnormal changes in feature correlations are mixed with normal fluctuations. Besides, according to the F1 scores, the performance of our algorithms are still not good enough to distinguish the healthy and faulty conditions even for the first two cases mentioned in this part. Such issues likely arises from the choice of indicators. Merging the correlation changes of all features into a single indicator, such as Euclidean distance, causes sub-indicators with large normal correlation changes to overshadow those with less significant changes. Consequently, even when the system enters an abnormal state, these abnormal correlation changes are not apparent, negatively impacting the algorithm's judgment. Thus, it would be more effective to process and analyze each sub-indicator representing the correlations separately.

6.2. Results: VARIMA + One-class SVM

The modified VARIMA + One-Class SVM approach starts with applying PCA to reduce the dataset's dimensionality, thereby enhancing computational efficiency. Next, the VARIMA model is utilized to capture the dynamics of the system's multivariate time series data, producing predictions for normal operating conditions. The residuals, which are the differences between the predicted and actual values, indicate any deviations from expected behavior. These residuals are then used to extract features for training a One-Class SVM model based on normal operating data. During deployment, new residual features are input into the trained One-Class SVM, which detects deviations from the normal range and effectively identifies and signals anomalies in the system.

Since the length of datasets is much slower that of the previous ones, here we select the first 1/40 of the healthy sequence of data to train VARIMA models and the following 1/20 of the rest of the healthy sequence to train One-class SVM models. And the VARIMA- and LSTM-related results of the following cases all share the same ratio of datasets division.

All results and performance indicators are listed in Appendix G. Although the accuracies exceed 90% for all six cases and the F1 scores for four out of the six cases are around 0.8 or even above 0.9, the F1 scores for the remaining two cases are around 0.6 or even nan (indicating no distinguishing ability).

There are two possible reasons for these issues:

1) For the most unique case among the six, case CDSEVENT0102023 (where the F1 score indicates that the code lost its ability to distinguish), it is evident that the values of all new indicators oscillate significantly. Since the One-class SVM algorithm cannot capture numerical correlations between different time points, these large fluctuations cause the threshold range for normal values in the model to become very large. Additionally, in this case, after dimensionality reduction by the PCA algorithm, the values of the new features returned to extremely small fluctuations after the actual failure occurred, causing the abnormal trend growth originally present in the raw data to disappear. This is one of the main reasons the code loses its distinguishing ability in this case.

2) By transforming the original features into new features via PCA, the original and independent characteristics of each feature are mixed and twisted, which confuses the algorithm.

6.3. Results without PCA: VARIMA + One-class SVM

The method of VARIMA + One-class SVM in this section is similar to that mentioned in section 6.2, but PCA is not applied to reduce dimensionality of datasets.

As discussed in the previous section, although the numerical performance indicators suggest better detection results for some cases compared to long-term performance results, there are still numerous isolated faulty notifications before the actual trip. Therefore, it is crucial to reduce the number of these unexpected notifications.

While PCA can reduce the dimensionality of datasets and increase computing speed, it distorts the original dataset's information. Additionally, the principal components may twist the deterministic characteristics of different original features. This research currently only considers vibration-related features, consisting of 8 tags, making it feasible to directly use the original data to train and test models.

All results and performance indicators are listed in Appendix H. It is evident that the code has significantly improved in all cases. Among the 6 cases, the F1 scores of the first 5 cases have increased to more than 0.96. For the 6th case (Case CDSEVENT0102023), which previously caused the code to completely lose its distinguishing ability, the F1 score, although still only 0.5111 due to large fluctuations in the original feature value, represents a significant improvement compared to nan. The detection results (Figure H.6) show that even the naked eye can distinguish normal and abnormal notifications based on the density of notifications.

However, it is undeniable that the large fluctuations in some feature values during the system's healthy state negatively impact the code's ability to distinguish. This thesis suggests two possible solutions to this problem:

1) Use a more complex algorithm model than the One-class SVM algorithm, which can only judge the value of a single time point. A more complex model could not only evaluate based on a single time point's deviation but also make corrections by linking the value relationships between previous and subsequent time points;

2) Normalize or standardize the data with large fluctuations to convert it into data with low fluctuations.

6.4. Results: LSTM + One-class SVM

The workflow of the modified LSTM + One-class SVM is mostly the same as that of VARIMA + One-class SVM mentioned in section 6.2. The only difference is that the regressor used in this section is changed from VARIMA to LSTM.

Appendix I shows the detection results and the indicators of performance for LSTM + One-class SVM with PCA. Since the new feature values are the same as those of VARIMA + One-class SVM with PCA, they are not shown in the appendix. Although the performance are generally better than that of VARIMA + One-class SVM with PCA, it takes much longer time to train the model. The main reason why LSTM takes longer to train the model than VARIMA is its complex model structure, computationally intensive training process, more data preprocessing steps, and dependence on hardware resources.

Besides, as shown in the detection results of case CDSEVENT0102023 (Figure I.6), the influence of data with large fluctuation in healthy conditions is still significant for the algorithm of LSTM + One-class SVM due to the characteristics of One-class SVM as mentioned in section 6.2.

6.5. Results without PCA: LSTM + One-class SVM

The LSTM + One-class SVM method described in this section is similar to the approach outlined in section 6.4, with the key difference being that PCA is not used to reduce the dimensionality of the datasets.

As shown in appendix J, both the accuracy and F1 scores show obvious increase compared with those of VARIMA + One-class SVM without PCA. And the detection result shown in the appendix indicates that the algorithm perform well in automatically filtering out the unrelated notifications and the influences caused by features with large fluctuations in values over time for the majority of all 6 cases. Such increase in performance mainly relies on the excellent ability to remember and extract (non)linear long-term characteristics of training data, which provides a more firm foundation of reference compared with that from VARIMA.

Generally speaking, although the results of LSTM + One-class SVM with/without PCA are clearly better than those of VARIMA + One-class SVM with/without PCA, the cost of time of the former one is much higher than that of the later one. And the results of LSTM + One-class SVM are obtained under the circumstance that the structures of LSTM models are kind of simple in this research, such as the number of hidden layers of LSTM being just 10. If we increase the complexity of the LSTM structures, more nonlinear characteristics can be extracted, further improving the performance of detection. However, the cost of increasing the complexity of the LSTM model is that the training time of the model increases exponentially. However, too long training time is not conducive to the application of the model in actual engineering environments. Therefore, in order to take advantage of LSTM in mining nonlinear characteristics of data, we need to further compress the time or improve the performance of the computer.

In general, the detection performance of LSTM + One-class SVM surpasses that of VARIMA + One-class SVM and PCA-based Prognosis. Since PCA distorts original statistical features and only vibration-related features are considered, PCA is unnecessary for LSTM- and VARIMA-based vibration monitoring approaches.

7 Evaluation and Improvement Suggestions

This chapter summarizes the analysis of the results mentioned in chapter 5 and chapter 6 and provide several suggestions for further improvements. Figure K.2 and Figure K.1 in Appendix K visualize the flowchart of how the proposed approaches are implemented, and/or how the approach can be implemented when it is sufficiently improved. Detailed introduction on the embedded content within these two figures is presented in the following sections.

7.1. Evaluation

7.1.1. General Performance Comparison

According to the numerical indicators, the detection performance of LSTM + One-class SVM is much better than that of VARIMA + One-class SVM and PCA-based Prognosis. Since only vibration-related features are considered in this research and PCA transformation twists the original statistical characteristics significantly, Principal Component Analysis is not necessary for both LSTM- and VARIMA-related approaches for vibration monitoring.

7.1.2. Computational Time

Although LSTM can provide best results by capturing more complex nonlinear relationships in data and provide most powerful detection capabilities among all three approaches, its computational time cost is considerable. In contrast, the VARIMA model has much lower computational complexity and faster training speed, and is suitable for detection tasks of (semi-)linear time series data.

7.1.3. Robustness in Dealing with Large Fluctuation in Values

As for robustness in dealing with features whose values oscillate significantly, LSTM + One-class SVM excels VARIMA + One-class SVM. PCA-based Prognosis cleverly avoids such issues by data normalization before calculating the correlations among features.

7.1.4. Indicator Selection

PCA-based Prognosis has great potential of improvement since it goes further into the inter-connections among features, which are more direct indicators to reveal the deep logic of different operational conditions compared with superficial analysis on feature values. However, the indicator of PCA-based Prognosis chosen in this research can mix the abnormal shifts of correlations with the normal ones, confusing the algorithm.

7.1.5. Capability to Identify Scheduled/Unrelated Shutdowns Automatically

It's still an unsolved question how to tell the scheduled/unrelated shutdowns from actual vibration-related shutdowns automatically. Thus, so far we can still insist on the strategy to restart algorithms after every shutdowns.

7.2. Improvement Suggestions

7.2.1. Functional Expansion from Detection to Prediction

During forehand investigation, we find that vibration problems can happen with different reasons, some of them not obviously shown. Thus, unless we have decisive indicators for prediction, the only methods we can now choose is to tune the parameters of detective models and let the models find out the hidden or ambiguous abnormalities ahead of actual trips, and this how prediction is fulfilled with detective models. So if our algorithms cannot perform well for detection, then it's not possible to generate stable and clear predictive results.

Now we have found the LSTM- and VARIMA-related methods can already fulfill good detection performance, and these two sets of algorithms can be regarded as the basis of predictive models. However, auxiliary codes should also be developed to decrease the number of false notifications ahead of the actual trips, especially for VARIMA + One-class SVM with/without PCA whose distinguishing ability is worse than LSTM + One-class SVM with/without PCA.

As for PCA-based Prognosis, since its detection performance is still not good enough, further improvement on detection is still needed, such as separate analysis on the correlation values. However, just as mentioned in the previous section, this approach focuses on much deeper statistical characteristics within data than the superficial analysis on the changes in data values only. Thus it might reveal more surprising results than the VARIMA- and LSTM-related algorithms in this research, since stable abnormal shifts ahead of actual trips are already shown in one or two cases.

To enable robust prediction, several additional elements are also required beyond enhancing detection performance.

- **Firstly**, comprehensive and high-quality data collection is essential, as diverse and representative datasets ensure the models can generalize well to various scenarios.
- **Secondly**, a refined methodology incorporating advanced feature engineering techniques can help capture more intricate patterns in the data.
- **Thirdly**, integration of domain knowledge can improve model interpretability and accuracy, guiding the selection of relevant features and the design of more effective algorithms.
- **Lastly**, continuous model validation and retraining with new data are necessary to adapt to changing conditions and maintain predictive accuracy over time.

7.2.2. Interpretation of False Notifications

After the rough detection results are obtained and filtering work is conducted, further interpretation for all the remaining false notifications is needed to determine the cause of issues. The most direct strategy is to send those notifications to onboard operators and let them conduct in-situ investigation to find out the cause of problems, which is labour-intensive and inefficient.

In order to transform the procedure of manual determination of problem sources into an automatic process, a set of cause/effect matrix and/or neural network can be developed to simulate the human brain's process of analyzing and judging notifications via computers. However, the work to simulate judgment progress of brain requires huge workload with large amounts of empirical data, including numerical values and records of each fault in text form. Long-term work in the future is needed to fulfill such functions.

8 Conclusion

8.1. Reflection on Research Questions

This section discusses how and to which extend this thesis answers the four research questions mentioned in Chapter 1.

- **Main Question:** How to predict or detect the abnormal events of vibration from data of water injection systems of FPSOs using Principal Component Analysis?

  The thesis demonstrates that while PCA-based approaches have potential, they currently underperform compared to LSTM + One-class SVM and VARIMA + One-class SVM in detection tasks.

- **Sub-Question 1:** How to effectively reduce dimensionality of the datasets of the target problem using Principal Component Analysis?

  This research finds that while PCA effectively reduces dimensionality, it negatively impacts the detection performance of LSTM- and VARIMA-related algorithms. This suggests a trade-off between dimensionality reduction and the preservation of critical statistical characteristics necessary for accurate detection.

- **Sub-Question 2:** How to predict or detect the abnormal events using the principal components of the datasets?

  This research shows that using principal components for prediction and detection is challenging and often less effective than other methods. The PCA-based Prognosis needs further refinement to improve its detection performance and realize its predictive potential.

- **Sub-Question 3:** What are the criteria for determining the optimal prediction/detection method?

  The optimal method is determined by its ability to accurately detect abnormalities, robustness to data fluctuations, and minimal false notifications. LSTM + One-class SVM meets these criteria best, followed by VARIMA + One-class SVM, with PCA-based Prognosis requiring further enhancement.

8.2. Summary

This thesis presents three sets of algorithms (PCA-based Prognosis, LSTM + One-class SVM and VARIMA + One-class SVM), analyzes their detective performance for vibration monitoring and discusses their potential for prediction.

As for the detective function, LSTM + One-class SVM excels both VARIMA + One-class SVM and PCA-based Prognosis in general performance. Meanwhile, this algorithm also shows strong robustness in tackling with disturbances caused by data with large fluctuation amplitudes. Besides, it is proven that dimensionality reduction via Principal Component Analysis exerts negative effects on the distinguishing ability of both LSTM- and VARIMA-related algorithms mentioned in this research.

As for predictive function, since no decisive indicators are found, predictive models can merely be developed in the basis of good detective models. Both LSTM- and VARIMA-related algorithms (without PCA) match the requirement. Although the performance of PCA-based Prognosis is still not satisfactory, its ability to directly mine the deep logic of equipment operating status also indicates that it may be able to obtain richer information than the other two codes, so its potential to transform into a predictive model cannot be ignored.

In conclusion, while LSTM + One-class SVM and VARIMA + One-class SVM are effective for detection and form a solid basis for predictive models, PCA-based methods, despite their current limitations, hold potential for deeper analysis and future predictive applications. Further research and improvement are necessary to fully leverage the capabilities of PCA in this context.

References

[1] C. Wang et al. “A Review of Vibration-Based Structural Health Monitoring with Special Emphasis on Composite Materials”. In: *Sensors* 18.1 (2018), p. 186.

[2] C. Zhang and Y. Qin. “A Deep Learning Approach to Gearbox Fault Diagnosis”. In: *IEEE Access* 8 (2020), pp. 10966–10975.

[3] D. J. Ewins. *Modal Testing: Theory, Practice, and Application*. Research Studies Press, 2000.

[4] S. Mallat. “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.7 (1989), pp. 674–693.

[5] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1999.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[8] C. M. Harris. *Shock and Vibration Handbook*. McGraw-Hill Professional, 2001.

[9] R. B. Randall. *Vibration-based Condition Monitoring: Industrial, Aerospace, and Automotive Applications*. John Wiley & Sons, 2011.

[10] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[11] J. Shlens. “A Tutorial on Principal Component Analysis”. In: *arXiv preprint arXiv:1404.1100* (2014).

[12] Mei-Ling Shyu et al. “A Novel Anomaly Detection Scheme Based on Principal Component Classifier”. In: 2003.

[13] Yule George Udny. “VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer’s sunspot numbers”. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 226 (1993), pp. 267–298. ISSN: 0264-3952.

[14] Pesi R Masani. *Norbert Wiener 1894–1964*. Vol. 5. Birkhäuser, 2012.

[15] Peter Whittle. “Hypothesis testing in time series analysis”. In: *Journal of the American Statistical Association* (1951), pp. 197–200.

[16] Gregory C. Reinsel George E. P. Box Gwilym M. Jenkins. *Time Series Analysis, Fourth Edition*. John Wiley & Sons, 2008.

[17] Cenk Temizel et al. “Production Forecasting in Shale Reservoirs Using LSTM Method in Deep Learning”. In: July 2020.

[18] GeeksforGeeks. *Understanding of LSTM Networks*. `https://www.geeksforgeeks.org/understanding-of-lstm-networks/`. Accessed on: 2024-07-09.

[19] Hyun Joon Shin, Dong-Hwan Eom, and Sung-Shick Kim. “One-class support vector machines—an application in machine fault detection and classification”. In: *Computers & Industrial Engineering* 48.2 (2005), pp. 395–408. ISSN: 0360-8352.

[20] Bernhard Schölkopf et al. “Support Vector Method for Novelty Detection”. In: vol. 12. Jan. 1999, pp. 582–588.

[21] H. Akaike. “A new look at the statistical model identification”. In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723.

[22] Kimberly L. Elmore and Michael B. Richman. “Euclidean Distance as a Similarity Metric for Principal Component Analysis”. In: *Monthly Weather Review* 129.3 (2001), pp. 540–549.

A
Framework of Water Injection Systems

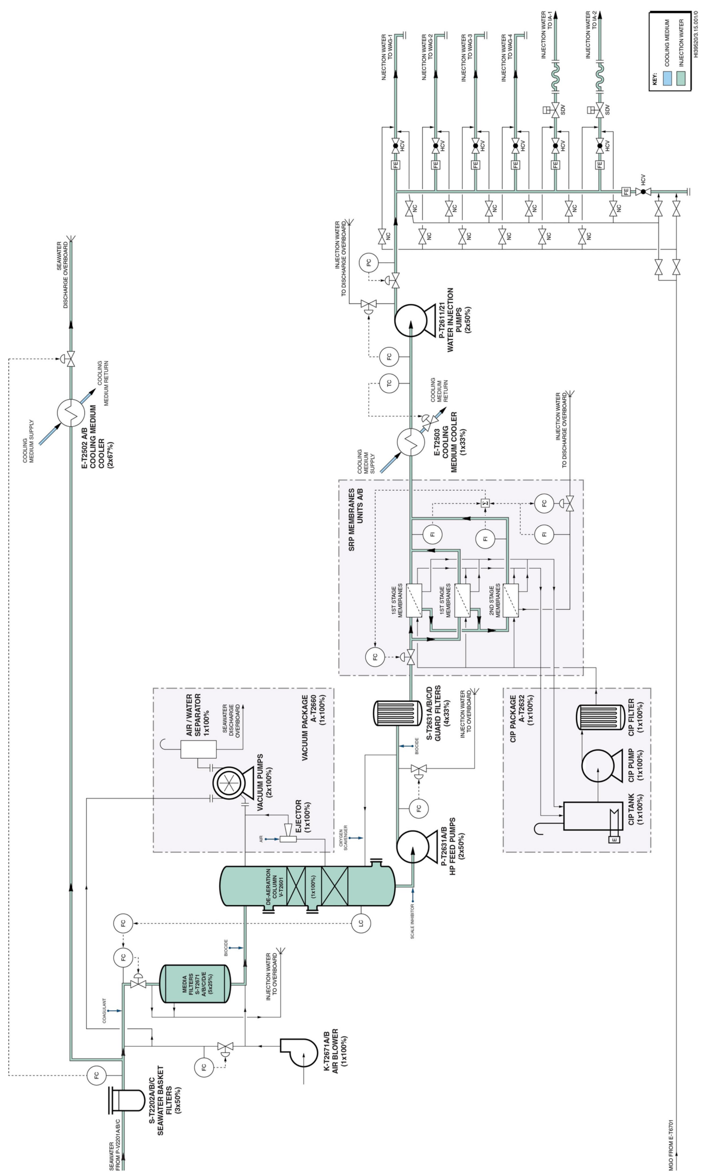

Figure A.1: Framework of Water Injection System

B

Water Injection System Tags of FPSO CDI/CDM/CDS

FPSO	Attribute Name	PI Tag Name
CDS	MLO Reservoir Level	CDS:FPSO:T26-LI-1873
CDS	MLO Reservoir Level	CDS:FPSO:T26-LI-1973
CDS	MLO Filter Differential Pressure	CDS:FPSO:T26-PDI-1874
CDS	MLO Filter Differential Pressure	CDS:FPSO:T26-PDI-1974
CDS	MLO Supply Pressure	CDS:FPSO:T26-PI-1871
CDS	Main MLO Pump Discharge Pressure	CDS:FPSO:T26-PI-1872
CDS	MLO Supply Pressure	CDS:FPSO:T26-PI-1971
CDS	Main MLO Pump Discharge Pressure	CDS:FPSO:T26-PI-1972
CDS	Pump Outboard Thrust Bearing Temperature	CDS:FPSO:T26-TI-1810
CDS	Pump Inboard Thrust Bearing Temperature	CDS:FPSO:T26-TI-1811
CDS	Pump NDE Radial Bearing Temperature	CDS:FPSO:T26-TI-1812
CDS	Pump DE Radial Bearing Temperature	CDS:FPSO:T26-TI-1815
CDS	Motor DE Radial Bearing Temperature	CDS:FPSO:T26-TI-1820
CDS	Winding Phase U2 Temperature	CDS:FPSO:T26-TI-1821
CDS	Winding Phase V2 Temperature	CDS:FPSO:T26-TI-1822
CDS	Winding Phase W2 Temperature	CDS:FPSO:T26-TI-1823
CDS	Winding Phase U1 Temperature	CDS:FPSO:T26-TI-1824
CDS	Winding Phase V1 Temperature	CDS:FPSO:T26-TI-1825
CDS	Winding Phase W1 Temperature	CDS:FPSO:T26-TI-1826
CDS	Motor NDE Radial Bearing Temperature	CDS:FPSO:T26-TI-1827
CDS	Cold Air Temperature	CDS:FPSO:T26-TI-1828
CDS	Cold Air Temperature 2	CDS:FPSO:T26-TI-1829
CDS	Hot Air Temperature	CDS:FPSO:T26-TI-1830
CDS	MLO Supply Temperature	CDS:FPSO:T26-TI-1875
CDS	MLO Reservoir Temperature	CDS:FPSO:T26-TI-1876
CDS	Pump Outboard Thrust Bearing Temperature	CDS:FPSO:T26-TI-1910
CDS	Pump Inboard Thrust Bearing Temperature	CDS:FPSO:T26-TI-1911
CDS	Pump NDE Radial Bearing Temperature	CDS:FPSO:T26-TI-1912
CDS	Pump DE Radial Bearing Temperature	CDS:FPSO:T26-TI-1915
CDS	Motor DE Radial Bearing Temperature	CDS:FPSO:T26-TI-1920
CDS	Winding Phase U2 Temperature	CDS:FPSO:T26-TI-1921
CDS	Winding Phase V2 Temperature	CDS:FPSO:T26-TI-1922
CDS	Winding Phase W2 Temperature	CDS:FPSO:T26-TI-1923
CDS	Winding Phase U1 Temperature	CDS:FPSO:T26-TI-1924
CDS	Winding Phase V1 Temperature	CDS:FPSO:T26-TI-1925

FPSO	Attribute Name	PI Tag Name
CDS	Winding Phase W1 Temperature	CDS:FPSO:T26-TI-1926
CDS	Motor NDE Radial Bearing Temperature	CDS:FPSO:T26-TI-1927
CDS	Cold Air Temperature	CDS:FPSO:T26-TI-1928
CDS	Cold Air Temperature 2	CDS:FPSO:T26-TI-1929
CDS	Hot Air Temperature	CDS:FPSO:T26-TI-1930
CDS	MLO Supply Temperature	CDS:FPSO:T26-TI-1975
CDS	MLO Reservoir Temperature	CDS:FPSO:T26-TI-1976
CDS	Pump NDE Radial Bearing vibration x	CDS:FPSO:T26-VXI-1811
CDS	Pump DE Radial Bearing vibration x	CDS:FPSO:T26-VXI-1815
CDS	Motor DE Radial Bearing Vibration x	CDS:FPSO:T26-VXI-1820
CDS	Motor NDE Radial Bearing Vibration x	CDS:FPSO:T26-VXI-1827
CDS	Pump NDE Radial Bearing vibration x	CDS:FPSO:T26-VXI-1911
CDS	Pump DE Radial Bearing vibration x	CDS:FPSO:T26-VXI-1915
CDS	Motor DE Radial Bearing Vibration x	CDS:FPSO:T26-VXI-1920
CDS	Motor NDE Radial Bearing Vibration x	CDS:FPSO:T26-VXI-1927
CDS	Pump NDE Radial Bearing vibration y	CDS:FPSO:T26-VYI-1812
CDS	Pump DE Radial Bearing vibration y	CDS:FPSO:T26-VYI-1816
CDS	Motor DE Radial Bearing Vibration y	CDS:FPSO:T26-VYI-1821
CDS	Motor NDE Radial Bearing Vibration y	CDS:FPSO:T26-VYI-1828
CDS	Pump NDE Radial Bearing vibration y	CDS:FPSO:T26-VYI-1912
CDS	Pump DE Radial Bearing vibration y	CDS:FPSO:T26-VYI-1916
CDS	Motor DE Radial Bearing Vibration y	CDS:FPSO:T26-VYI-1921
CDS	Motor NDE Radial Bearing Vibration y	CDS:FPSO:T26-VYI-1928
CDS	Pump Shaft Axial Displacement	CDS:FPSO:T26-ZE-1811
CDS	Pump Shaft Axial Displacement 2	CDS:FPSO:T26-ZE-1812
CDS	Pump Shaft Axial Displacement	CDS:FPSO:T26-ZE-1911
CDS	Pump Shaft Axial Displacement 2	CDS:FPSO:T26-ZE-1912
CDM	MLO Reservoir Level	CDM:FPSO:T26-LSI-1873
CDM	MLO Reservoir Level	CDM:FPSO:T26-LSI-1973
CDM	MLO Filter Differential Pressure	CDM:FPSO:T26-PDI-1874
CDM	MLO Filter Differential Pressure	CDM:FPSO:T26-PDI-1974
CDM	Main MLO Pump Discharge Pressure	CDM:FPSO:T26-PI-1872
CDM	Main MLO Pump Discharge Pressure	CDM:FPSO:T26-PI-1972
CDM	Pump DE Radial Bearing Temperature	CDM:FPSO:T26-TI-1815
CDM	Motor DE Radial Bearing Temperature	CDM:FPSO:T26-TI-1820
CDM	Winding Phase U2 Temperature	CDM:FPSO:T26-TI-1821
CDM	Winding Phase V2 Temperature	CDM:FPSO:T26-TI-1822
CDM	Winding Phase W2 Temperature	CDM:FPSO:T26-TI-1823
CDM	Winding Phase U1 Temperature	CDM:FPSO:T26-TI-1824
CDM	Winding Phase V1 Temperature	CDM:FPSO:T26-TI-1825
CDM	Winding Phase W1 Temperature	CDM:FPSO:T26-TI-1826
CDM	Motor NDE Radial Bearing Temperature	CDM:FPSO:T26-TI-1827
CDM	Cold Air Temperature	CDM:FPSO:T26-TI-1828
CDM	Cold Air Temperature 2	CDM:FPSO:T26-TI-1829
CDM	Hot Air Temperature	CDM:FPSO:T26-TI-1830
CDM	Motor DE Radial Bearing Temperature	CDM:FPSO:T26-TI-1920
CDM	Winding Phase U2 Temperature	CDM:FPSO:T26-TI-1921
CDM	Winding Phase V2 Temperature	CDM:FPSO:T26-TI-1922
CDM	Winding Phase W2 Temperature	CDM:FPSO:T26-TI-1923
CDM	Winding Phase U1 Temperature	CDM:FPSO:T26-TI-1924
CDM	Winding Phase V1 Temperature	CDM:FPSO:T26-TI-1925
CDM	Winding Phase W1 Temperature	CDM:FPSO:T26-TI-1926
CDM	Motor NDE Radial Bearing Temperature	CDM:FPSO:T26-TI-1927
CDM	Cold Air Temperature	CDM:FPSO:T26-TI-1928

FPSO	Attribute Name	PI Tag Name
CDM	Cold Air Temperature 2	CDM:FPSO:T26-TI-1929
CDM	Hot Air Temperature	CDM:FPSO:T26-TI-1930
CDM	Pump DE Radial Bearing Temperature	CDM:FPSO:T26-TI-1915
CDM	Motor DE Radial Bearing Vibration x	CDM:FPSO:T26-VXI-1820
CDM	Motor DE Radial Bearing Vibration x	CDM:FPSO:T26-VXI-1920
CDM	Motor DE Radial Bearing Vibration y	CDM:FPSO:T26-VYI-1821
CDM	Motor DE Radial Bearing Vibration y	CDM:FPSO:T26-VYI-1921
CDM	Motor NDE Radial Bearing Vibration x	CDM:FPSO:T26-VXI-1827
CDM	Motor NDE Radial Bearing Vibration x	CDM:FPSO:T26-VXI-1927
CDM	Motor NDE Radial Bearing Vibration y	CDM:FPSO:T26-VYI-1828
CDM	Motor NDE Radial Bearing Vibration y	CDM:FPSO:T26-VYI-1928
CDM	Pump DE Radial Bearing vibration x	CDM:FPSO:T26-VXI-1815
CDM	Pump DE Radial Bearing vibration x	CDM:FPSO:T26-VXI-1915
CDM	MLO Reservoir Temperature	CDM:FPSO:T26-TI-1876
CDM	MLO Reservoir Temperature	CDM:FPSO:T26-TI-1976
CDM	MLO Supply Pressure	CDM:FPSO:T26-PI-1871
CDM	MLO Supply Pressure	CDM:FPSO:T26-PI-1971
CDM	MLO Supply Temperature	CDM:FPSO:T26-TI-1875
CDM	MLO Supply Temperature	CDM:FPSO:T26-TI-1975
CDM	Pump DE Radial Bearing vibration y	CDM:FPSO:T26-VYI-1816
CDM	Pump DE Radial Bearing vibration y	CDM:FPSO:T26-VYI-1916
CDM	Pump Inboard Thrust Bearing Temperature	CDM:FPSO:T26-TI-1811
CDM	Pump Inboard Thrust Bearing Temperature	CDM:FPSO:T26-TI-1911
CDM	Pump NDE Radial Bearing Temperature	CDM:FPSO:T26-TI-1812
CDM	Pump NDE Radial Bearing Temperature	CDM:FPSO:T26-TI-1912
CDM	Pump NDE Radial Bearing vibration x	CDM:FPSO:T26-VXI-1811
CDM	Pump NDE Radial Bearing vibration x	CDM:FPSO:T26-VXI-1911
CDM	Pump NDE Radial Bearing vibration y	CDM:FPSO:T26-VYI-1812
CDM	Pump NDE Radial Bearing vibration y	CDM:FPSO:T26-VYI-1912
CDM	Pump Outboard Thrust Bearing Temperature	CDM:FPSO:T26-TI-1810
CDM	Pump Outboard Thrust Bearing Temperature	CDM:FPSO:T26-TI-1910
CDM	Pump Shaft Axial Displacement	CDM:FPSO:T26-ZI-1811
CDM	Pump Shaft Axial Displacement 2	CDM:FPSO:T26-ZI-1812
CDM	Pump Shaft Axial Displacement	CDM:FPSO:T26-ZI-1911
CDM	Pump Shaft Axial Displacement 2	CDM:FPSO:T26-ZI-1912
CDI	MLO Reservoir Level	CDI:FPSO:T26-LI-1873
CDI	MLO Reservoir Level	CDI:FPSO:T26-LI-1973
CDI	Main MLO Pump Discharge Pressure	CDI:FPSO:T26-PI-1872
CDI	MLO Filter Differential Pressure	CDI:FPSO:T26-PDI-1874
CDI	MLO Filter Differential Pressure	CDI:FPSO:T26-PDI-1974
CDI	MLO Supply Pressure	CDI:FPSO:T26-PI-1871
CDI	MLO Supply Pressure	CDI:FPSO:T26-PI-1971
CDI	Main MLO Pump Discharge Pressure	CDI:FPSO:T26-PI-1972
CDI	Pump Outboard Thrust Bearing Temperature	CDI:FPSO:T26-TI-1810
CDI	Pump Inboard Thrust Bearing Temperature	CDI:FPSO:T26-TI-1811
CDI	Pump NDE Radial Bearing Temperature	CDI:FPSO:T26-TI-1812
CDI	Pump DE Radial Bearing Temperature	CDI:FPSO:T26-TI-1815
CDI	Motor DE Radial Bearing Temperature	CDI:FPSO:T26-TI-1820
CDI	Motor NDE Radial Bearing Temperature	CDI:FPSO:T26-TI-1827
CDI	Cold Air Temperature	CDI:FPSO:T26-TI-1828
CDI	Cold Air Temperature 2	CDI:FPSO:T26-TI-1829
CDI	Hot Air Temperature	CDI:FPSO:T26-TI-1830
CDI	MLO Supply Temperature	CDI:FPSO:T26-TI-1875
CDI	MLO Reservoir Temperature	CDI:FPSO:T26-TI-1876

FPSO	Attribute Name	PI Tag Name
CDI	Pump Outboard Thrust Bearing Temperature	CDI:FPSO:T26-TI-1910
CDI	Pump Inboard Thrust Bearing Temperature	CDI:FPSO:T26-TI-1911
CDI	Pump NDE Radial Bearing Temperature	CDI:FPSO:T26-TI-1912
CDI	Pump DE Radial Bearing Temperature	CDI:FPSO:T26-TI-1915
CDI	Motor DE Radial Bearing Temperature	CDI:FPSO:T26-TI-1920
CDI	MLO Supply Temperature	CDI:FPSO:T26-TI-1975
CDI	MLO Reservoir Temperature	CDI:FPSO:T26-TI-1976
CDI	Motor NDE Radial Bearing Temperature	CDI:FPSO:T26-TI-1927
CDI	Cold Air Temperature	CDI:FPSO:T26-TI-1928
CDI	Cold Air Temperature 2	CDI:FPSO:T26-TI-1929
CDI	Hot Air Temperature	CDI:FPSO:T26-TI-1930
CDI	Pump NDE Radial Bearing vibration x	CDI:FPSO:T26-VXI-1811
CDI	Pump NDE Radial Bearing vibration y	CDI:FPSO:T26-VYI-1812
CDI	Pump DE Radial Bearing vibration x	CDI:FPSO:T26-VXI-1815
CDI	Motor DE Radial Bearing Vibration x	CDI:FPSO:T26-VXI-1820
CDI	Motor NDE Radial Bearing Vibration x	CDI:FPSO:T26-VXI-1827
CDI	Pump DE Radial Bearing vibration x	CDI:FPSO:T26-VXI-1915
CDI	Motor DE Radial Bearing Vibration x	CDI:FPSO:T26-VXI-1920
CDI	Motor NDE Radial Bearing Vibration x	CDI:FPSO:T26-VXI-1927
CDI	Pump DE Radial Bearing vibration y	CDI:FPSO:T26-VYI-1816
CDI	Motor DE Radial Bearing Vibration y	CDI:FPSO:T26-VYI-1821
CDI	Motor NDE Radial Bearing Vibration y	CDI:FPSO:T26-VYI-1828
CDI	Pump NDE Radial Bearing vibration x	CDI:FPSO:T26-VXI-1911
CDI	Pump NDE Radial Bearing vibration y	CDI:FPSO:T26-VYI-1912
CDI	Pump DE Radial Bearing vibration y	CDI:FPSO:T26-VYI-1916
CDI	Motor DE Radial Bearing Vibration y	CDI:FPSO:T26-VYI-1921
CDI	Motor NDE Radial Bearing Vibration y	CDI:FPSO:T26-VYI-1928
CDI	Pump Shaft Axial Displacement	CDI:FPSO:T26-ZE-1811
CDI	Pump Shaft Axial Displacement 2	CDI:FPSO:T26-ZE-1812
CDI	Pump Shaft Axial Displacement	CDI:FPSO:T26-ZE-1911
CDI	Pump Shaft Axial Displacement 2	CDI:FPSO:T26-ZE-1912

C

Tags of Water Injection Pump A/B

CDS	CDM	CDI
CDS:FPSO:PM-T2611	CDM:FPSO:PM-T2611	CDI:FPSO:PM-T2611
CDS:FPSO:PM-T2621	CDM:FPSO:PM-T2621	CDI:FPSO:PM-T2621
CDS:FPSO:T26-FI-1101	CDM:FPSO:T26-FI-1101	CDI:FPSO:T26-FI-1101
CDS:FPSO:T26-FI-1201	CDM:FPSO:T26-FI-1201	CDI:FPSO:T26-FI-1201
CDS:FPSO:T26-FSI-1102	CDM:FPSO:T26-FSI-1102	CDI:FPSO:T26-FSI-1102
CDS:FPSO:T26-FSI-1202	CDM:FPSO:T26-FSI-1202	CDI:FPSO:T26-FSI-1202
CDS:FPSO:T26-KE-1910	CDM:FPSO:T26-KE-1910	CDI:FPSO:T26-KE-1910
CDS:FPSO:T26-PDI-1105	CDM:FPSO:T26-PDI-1105	CDI:FPSO:T26-PDI-1105
CDS:FPSO:T26-PDI-1205	CDM:FPSO:T26-PDI-1205	CDI:FPSO:T26-PDI-1205
CDS:FPSO:T26-PSI-1101	CDM:FPSO:T26-PSI-1101	CDI:FPSO:T26-PSI-1101
CDS:FPSO:T26-PSI-1102	CDM:FPSO:T26-PSI-1102	CDI:FPSO:T26-PSI-1102
CDS:FPSO:T26-PSI-1201	CDM:FPSO:T26-PSI-1201	CDI:FPSO:T26-PSI-1201
CDS:FPSO:T26-PSI-1202	CDM:FPSO:T26-PSI-1202	CDI:FPSO:T26-PSI-1202
CDS:FPSO:T26-TI-1810	CDM:FPSO:T26-TI-1810	CDI:FPSO:T26-TI-1810
CDS:FPSO:T26-TI-1811	CDM:FPSO:T26-TI-1811	CDI:FPSO:T26-TI-1811
CDS:FPSO:T26-TI-1812	CDM:FPSO:T26-TI-1812	CDI:FPSO:T26-TI-1812
CDS:FPSO:T26-TI-1815	CDM:FPSO:T26-TI-1815	CDI:FPSO:T26-TI-1815
CDS:FPSO:T26-TI-1820	CDM:FPSO:T26-TI-1820	CDI:FPSO:T26-TI-1820
CDS:FPSO:T26-TI-1827	CDM:FPSO:T26-TI-1827	CDI:FPSO:T26-TI-1827
CDS:FPSO:T26-TI-1828	CDM:FPSO:T26-TI-1828	CDI:FPSO:T26-TI-1828
CDS:FPSO:T26-TI-1829	CDM:FPSO:T26-TI-1829	CDI:FPSO:T26-TI-1829
CDS:FPSO:T26-TI-1830	CDM:FPSO:T26-TI-1830	CDI:FPSO:T26-TI-1830
CDS:FPSO:T26-TI-1910	CDM:FPSO:T26-TI-1910	CDI:FPSO:T26-TI-1910
CDS:FPSO:T26-TI-1911	CDM:FPSO:T26-TI-1911	CDI:FPSO:T26-TI-1911
CDS:FPSO:T26-TI-1912	CDM:FPSO:T26-TI-1912	CDI:FPSO:T26-TI-1912
CDS:FPSO:T26-TI-1915	CDM:FPSO:T26-TI-1915	CDI:FPSO:T26-TI-1915
CDS:FPSO:T26-TI-1920	CDM:FPSO:T26-TI-1920	CDI:FPSO:T26-TI-1920
CDS:FPSO:T26-TI-1927	CDM:FPSO:T26-TI-1927	CDI:FPSO:T26-TI-1927
CDS:FPSO:T26-TI-1928	CDM:FPSO:T26-TI-1928	CDI:FPSO:T26-TI-1928
CDS:FPSO:T26-TI-1929	CDM:FPSO:T26-TI-1929	CDI:FPSO:T26-TI-1929
CDS:FPSO:T26-TI-1930	CDM:FPSO:T26-TI-1930	CDI:FPSO:T26-TI-1930
CDS:FPSO:T26-VXI-1811	CDM:FPSO:T26-VXI-1811	CDI:FPSO:T26-VXI-1811
CDS:FPSO:T26-VXI-1815	CDM:FPSO:T26-VXI-1815	CDI:FPSO:T26-VXI-1815
CDS:FPSO:T26-VXI-1820	CDM:FPSO:T26-VXI-1820	CDI:FPSO:T26-VXI-1820
CDS:FPSO:T26-VXI-1827	CDM:FPSO:T26-VXI-1827	CDI:FPSO:T26-VXI-1827
CDS:FPSO:T26-VXI-1911	CDM:FPSO:T26-VXI-1911	CDI:FPSO:T26-VXI-1911
CDS:FPSO:T26-VXI-1915	CDM:FPSO:T26-VXI-1915	CDI:FPSO:T26-VXI-1915
CDS:FPSO:T26-VXI-1920	CDM:FPSO:T26-VXI-1920	CDI:FPSO:T26-VXI-1920

CDS	CDM	CDI
CDS:FPSO:T26-VXI-1927	CDM:FPSO:T26-VXI-1927	CDI:FPSO:T26-VXI-1927
CDS:FPSO:T26-VYI-1812	CDM:FPSO:T26-VYI-1812	CDI:FPSO:T26-VYI-1812
CDS:FPSO:T26-VYI-1816	CDM:FPSO:T26-VYI-1816	CDI:FPSO:T26-VYI-1816
CDS:FPSO:T26-VYI-1821	CDM:FPSO:T26-VYI-1821	CDI:FPSO:T26-VYI-1821
CDS:FPSO:T26-VYI-1828	CDM:FPSO:T26-VYI-1828	CDI:FPSO:T26-VYI-1828
CDS:FPSO:T26-VYI-1912	CDM:FPSO:T26-VYI-1912	CDI:FPSO:T26-VYI-1912
CDS:FPSO:T26-VYI-1916	CDM:FPSO:T26-VYI-1916	CDI:FPSO:T26-VYI-1916
CDS:FPSO:T26-VYI-1921	CDM:FPSO:T26-VYI-1921	CDI:FPSO:T26-VYI-1921
CDS:FPSO:T26-VYI-1928	CDM:FPSO:T26-VYI-1928	CDI:FPSO:T26-VYI-1928
CDS:FPSO:T26-ZE-1811	CDM:FPSO:T26-ZE-1811	CDI:FPSO:T26-ZE-1811
CDS:FPSO:T26-ZE-1812	CDM:FPSO:T26-ZE-1812	CDI:FPSO:T26-ZE-1812
CDS:FPSO:T26-ZE-1911	CDM:FPSO:T26-ZE-1911	CDI:FPSO:T26-ZE-1911
CDS:FPSO:T26-ZE-1912	CDM:FPSO:T26-ZE-1912	CDI:FPSO:T26-ZE-1912
CDS:FPSO:T26-TI-1876	CDM:FPSO:T26-TI-1876	CDI:FPSO:T26-TI-1876
CDS:FPSO:T26-PDI-1874	CDM:FPSO:T26-PDI-1874	CDI:FPSO:T26-PDI-1874
CDS:FPSO:T26-PI-1871	CDM:FPSO:T26-PI-1871	CDI:FPSO:T26-PI-1871
CDS:FPSO:T26-TI-1875	CDM:FPSO:T26-TI-1875	CDI:FPSO:T26-TI-1875
CDS:FPSO:T26-PI-1872	CDM:FPSO:T26-PI-1872	CDI:FPSO:T26-PI-1872
CDS:FPSO:T26-LI-1873	CDM:FPSO:T26-LI-1873	CDI:FPSO:T26-LI-1873
CDS:FPSO:T26-TI-1976	CDM:FPSO:T26-TI-1976	CDI:FPSO:T26-TI-1976
CDS:FPSO:T26-PDI-1974	CDM:FPSO:T26-PDI-1974	CDI:FPSO:T26-PDI-1974
CDS:FPSO:T26-PI-1971	CDM:FPSO:T26-PI-1971	CDI:FPSO:T26-PI-1971
CDS:FPSO:T26-TI-1975	CDM:FPSO:T26-TI-1975	CDI:FPSO:T26-TI-1975
CDS:FPSO:T26-PI-1972	CDM:FPSO:T26-PI-1972	CDI:FPSO:T26-PI-1972
CDS:FPSO:T26-LI-1973	CDM:FPSO:T26-LI-1973	CDI:FPSO:T26-LI-1973

D

Remaining Tags of Water Injection Pump A/B

CDS	CDM	CDI
CDS:FPSO:T26-VXI-1811	CDM:FPSO:T26-VXI-1811	CDI:FPSO:T26-VXI-1811
CDS:FPSO:T26-VXI-1815	CDM:FPSO:T26-VXI-1815	CDI:FPSO:T26-VXI-1815
CDS:FPSO:T26-VXI-1820	CDM:FPSO:T26-VXI-1820	CDI:FPSO:T26-VXI-1820
CDS:FPSO:T26-VXI-1827	CDM:FPSO:T26-VXI-1827	CDI:FPSO:T26-VXI-1827
CDS:FPSO:T26-VXI-1911	CDM:FPSO:T26-VXI-1911	CDI:FPSO:T26-VXI-1911
CDS:FPSO:T26-VXI-1915	CDM:FPSO:T26-VXI-1915	CDI:FPSO:T26-VXI-1915
CDS:FPSO:T26-VXI-1920	CDM:FPSO:T26-VXI-1920	CDI:FPSO:T26-VXI-1920
CDS:FPSO:T26-VXI-1927	CDM:FPSO:T26-VXI-1927	CDI:FPSO:T26-VXI-1927
CDS:FPSO:T26-VYI-1812	CDM:FPSO:T26-VYI-1812	CDI:FPSO:T26-VYI-1812
CDS:FPSO:T26-VYI-1816	CDM:FPSO:T26-VYI-1816	CDI:FPSO:T26-VYI-1816
CDS:FPSO:T26-VYI-1821	CDM:FPSO:T26-VYI-1821	CDI:FPSO:T26-VYI-1821
CDS:FPSO:T26-VYI-1828	CDM:FPSO:T26-VYI-1828	CDI:FPSO:T26-VYI-1828
CDS:FPSO:T26-VYI-1912	CDM:FPSO:T26-VYI-1912	CDI:FPSO:T26-VYI-1912
CDS:FPSO:T26-VYI-1916	CDM:FPSO:T26-VYI-1916	CDI:FPSO:T26-VYI-1916
CDS:FPSO:T26-VYI-1921	CDM:FPSO:T26-VYI-1921	CDI:FPSO:T26-VYI-1921
CDS:FPSO:T26-VYI-1928	CDM:FPSO:T26-VYI-1928	CDI:FPSO:T26-VYI-1928

E

Tags of HP Feed Pump A/B

CDS	CDM	CDI
CDS:FPSO:PM-T2631A	CDM:FPSO:PM-T2631A	CDI:FPSO:PM-T2631A
CDS:FPSO:PM-T2631B	CDM:FPSO:PM-T2631B	CDI:FPSO:PM-T2631B
CDS:FPSO:T26-FIC-0681	CDM:FPSO:T26-FIC-0681	CDI:FPSO:T26-FIC-0681
CDS:FPSO:T26-PSI-0612	CDM:FPSO:T26-PSI-0612	CDI:FPSO:T26-PSI-0612
CDS:FPSO:T26-PSI-0621	CDM:FPSO:T26-PSI-0621	CDI:FPSO:T26-PSI-0621
CDS:FPSO:T26-PSI-0622	CDM:FPSO:T26-PSI-0622	CDI:FPSO:T26-PSI-0622
CDS:FPSO:T26-TI-0683	CDM:FPSO:T26-TI-0683	CDI:FPSO:T26-TI-0683
CDS:FPSO:T26-TSI-1627	CDM:FPSO:T26-TSI-1627	CDI:FPSO:T26-TSI-1627
CDS:FPSO:T26-TSI-1710	CDM:FPSO:T26-TSI-1710	CDI:FPSO:T26-TSI-1710
CDS:FPSO:T26-TSI-1711	CDM:FPSO:T26-TSI-1711	CDI:FPSO:T26-TSI-1711
CDS:FPSO:T26-TSI-1720	CDM:FPSO:T26-TSI-1720	CDI:FPSO:T26-TSI-1720
CDS:FPSO:T26-TSI-1727	CDM:FPSO:T26-TSI-1727	CDI:FPSO:T26-TSI-1727
CDS:FPSO:T26-VI-1610	CDM:FPSO:T26-VI-1610	CDI:FPSO:T26-VI-1610
CDS:FPSO:T26-VI-1710	CDM:FPSO:T26-VI-1710	CDI:FPSO:T26-VI-1710
CDS:FPSO:T26-VXI-1608	CDM:FPSO:T26-VXI-1608	CDI:FPSO:T26-VXI-1608
CDS:FPSO:T26-VXI-1615	CDM:FPSO:T26-VXI-1615	CDI:FPSO:T26-VXI-1615
CDS:FPSO:T26-VXI-1706	CDM:FPSO:T26-VXI-1706	CDI:FPSO:T26-VXI-1706
CDS:FPSO:T26-VXI-1708	CDM:FPSO:T26-VXI-1708	CDI:FPSO:T26-VXI-1708
CDS:FPSO:T26-VXI-1713	CDM:FPSO:T26-VXI-1713	CDI:FPSO:T26-VXI-1713
CDS:FPSO:T26-VYI-1607	CDM:FPSO:T26-VYI-1607	CDI:FPSO:T26-VYI-1607
CDS:FPSO:T26-VYI-1612	CDM:FPSO:T26-VYI-1612	CDI:FPSO:T26-VYI-1612
CDS:FPSO:T26-VYI-1707	CDM:FPSO:T26-VYI-1707	CDI:FPSO:T26-VYI-1707
CDS:FPSO:T26-VYI-1709	CDM:FPSO:T26-VYI-1709	CDI:FPSO:T26-VYI-1709
CDS:FPSO:T26-VYI-1712	CDM:FPSO:T26-VYI-1712	CDI:FPSO:T26-VYI-1712
CDS:FPSO:T26-VYI-1716	CDM:FPSO:T26-VYI-1716	CDI:FPSO:T26-VYI-1716

F

Remaining Tags of HP Feed Pump A/B

CDS	CDM	CDI
CDS:FPSO:T26-VI-1610	CDM:FPSO:T26-VI-1610	CDI:FPSO:T26-VI-1610
CDS:FPSO:T26-VI-1710	CDM:FPSO:T26-VI-1710	CDI:FPSO:T26-VI-1710
CDS:FPSO:T26-VXI-1608	CDM:FPSO:T26-VXI-1608	CDI:FPSO:T26-VXI-1608
CDS:FPSO:T26-VXI-1615	CDM:FPSO:T26-VXI-1615	CDI:FPSO:T26-VXI-1615
CDS:FPSO:T26-VXI-1706	CDM:FPSO:T26-VXI-1706	CDI:FPSO:T26-VXI-1706
CDS:FPSO:T26-VXI-1708	CDM:FPSO:T26-VXI-1708	CDI:FPSO:T26-VXI-1708
CDS:FPSO:T26-VXI-1713	CDM:FPSO:T26-VXI-1713	CDI:FPSO:T26-VXI-1713
CDS:FPSO:T26-VYI-1607	CDM:FPSO:T26-VYI-1607	CDI:FPSO:T26-VYI-1607
CDS:FPSO:T26-VYI-1612	CDM:FPSO:T26-VYI-1612	CDI:FPSO:T26-VYI-1612
CDS:FPSO:T26-VYI-1707	CDM:FPSO:T26-VYI-1707	CDI:FPSO:T26-VYI-1707
CDS:FPSO:T26-VYI-1709	CDM:FPSO:T26-VYI-1709	CDI:FPSO:T26-VYI-1709
CDS:FPSO:T26-VYI-1712	CDM:FPSO:T26-VYI-1712	CDI:FPSO:T26-VYI-1712
CDS:FPSO:T26-VYI-1716	CDM:FPSO:T26-VYI-1716	CDI:FPSO:T26-VYI-1716

G

Short-term Detection Results: VARIMA + One-class SVM with PCA

Table G.1: Performance of VARIMA + One-class SVM with PCA for Different Datasets

Dataset	Accuracy	Precision	Recall	F1
CDIEVENT0012021	0.8810	0.5768	0.8335	0.6009
CDMEVENT0012022	0.9888	0.9878	0.9458	0.9656
CDMEVENT0062023	0.9085	0.7207	0.9505	0.7802
CDSEVENT0052022	0.9381	0.8331	0.9000	0.8561
CDSEVENT0072022	0.9910	0.9747	0.8665	0.9132
CDSEVENT0102023	0.9588	0.4794	0.4999	nan
Dataset	Model Training/s	Abnormality Detection/s		
CDIEVENT0012021	504.9708	614.9422		
CDMEVENT0012022	565.0197	113.0982		
CDMEVENT0062023	707.3840	469.6973		
CDSEVENT0052022	634.3560	423.8700		
CDSEVENT0072022	788.2499	277.5725		
CDSEVENT0102023	360.6036	180.0022		
Dataset	Detection Steps	Detection per Time Step		
CDIEVENT0012021	363007	0.001694023		
CDMEVENT0012022	226898	0.000498454		
CDMEVENT0062023	500409	0.000938627		
CDSEVENT0052022	441913	0.000959170		
CDSEVENT0072022	258778	0.001072628		
CDSEVENT0102023	202179	0.000890311		

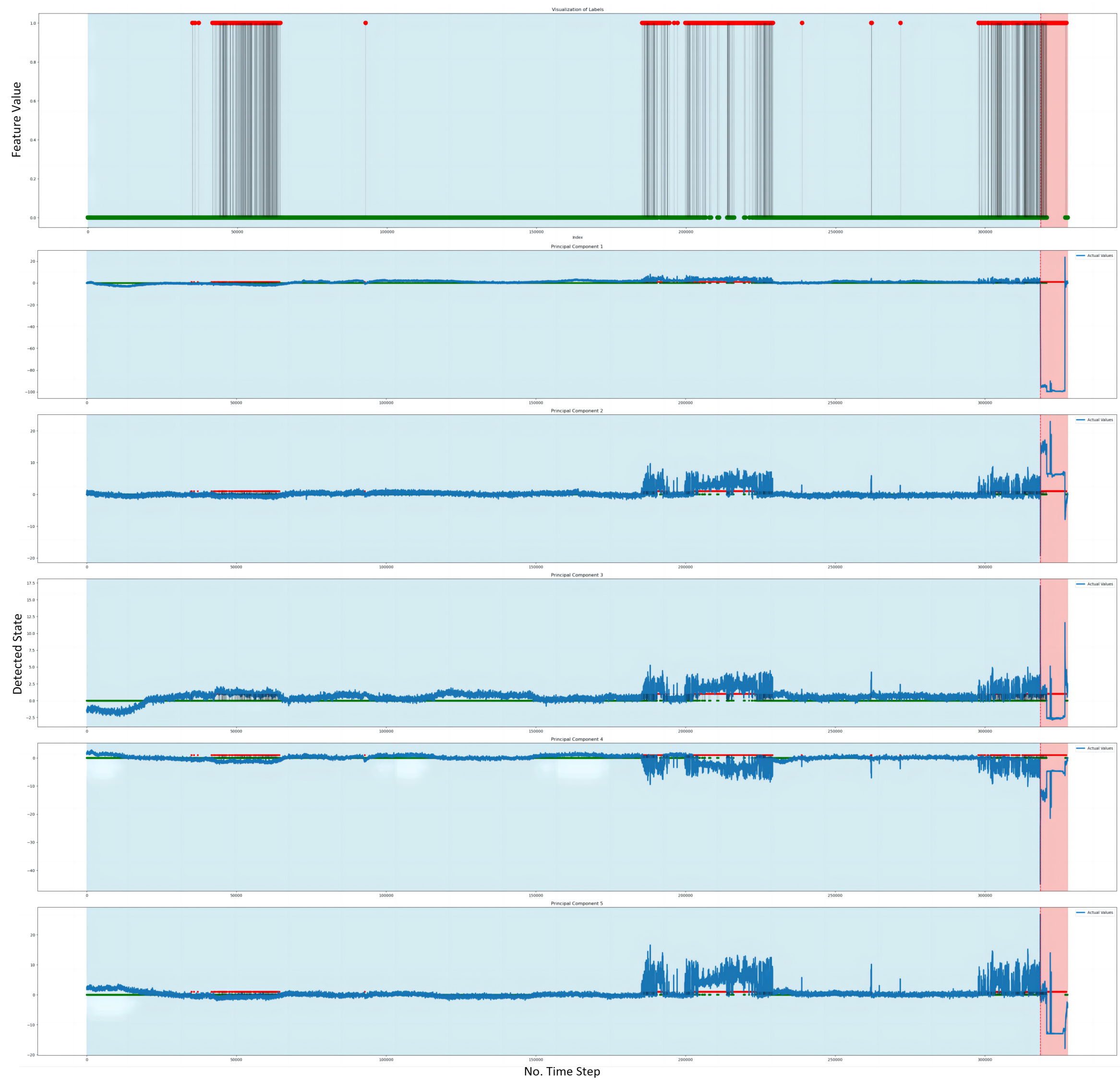

Figure G.1: CDIEVENT0012021: Short-term Detection Results for VARIMA + One-class SVM with PCA and New Feature Values

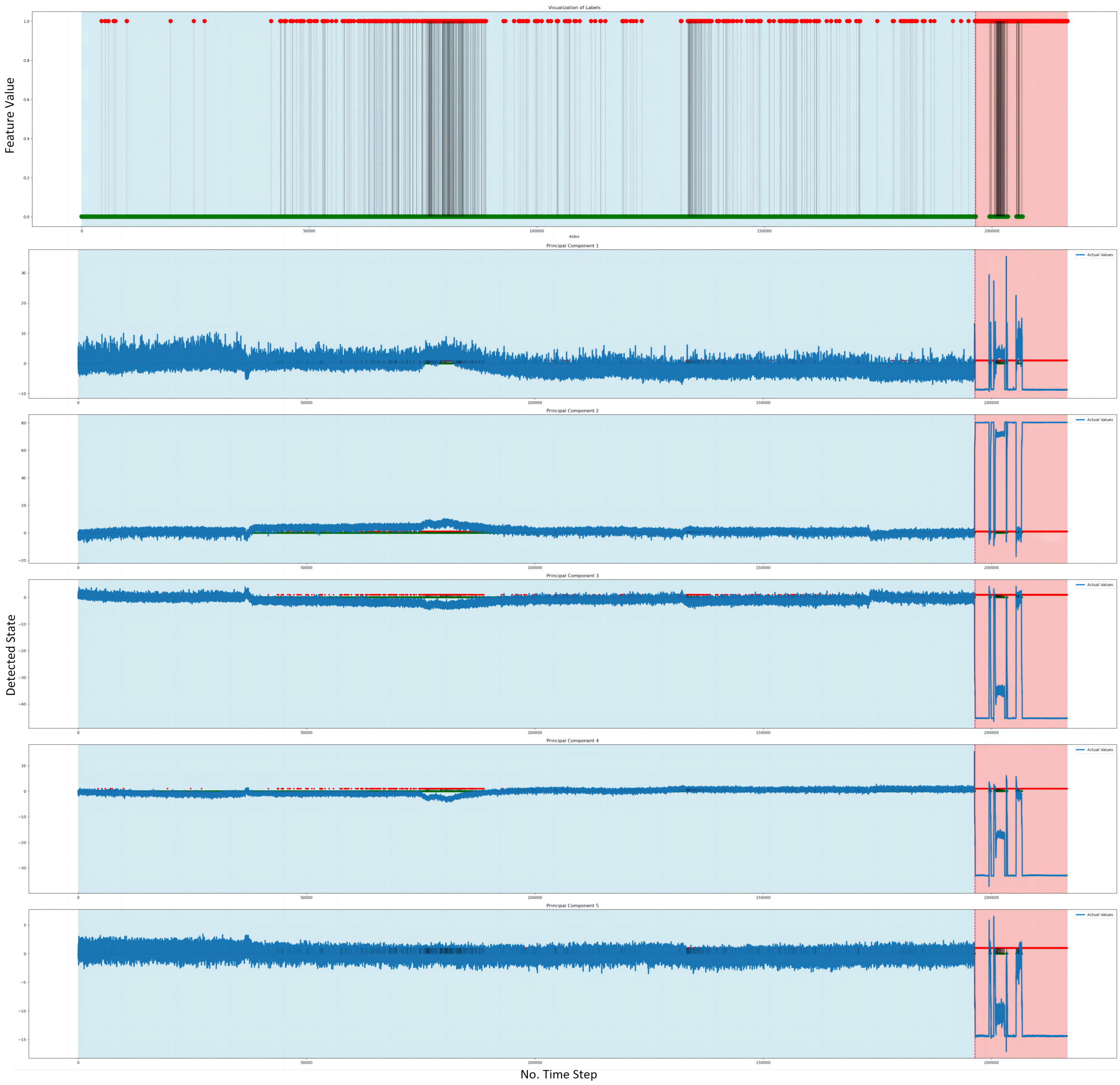

Figure G.2: CDMEVENT0012022: Short-term Detection Results for VARIMA + One-class SVM with PCA and New Feature Values

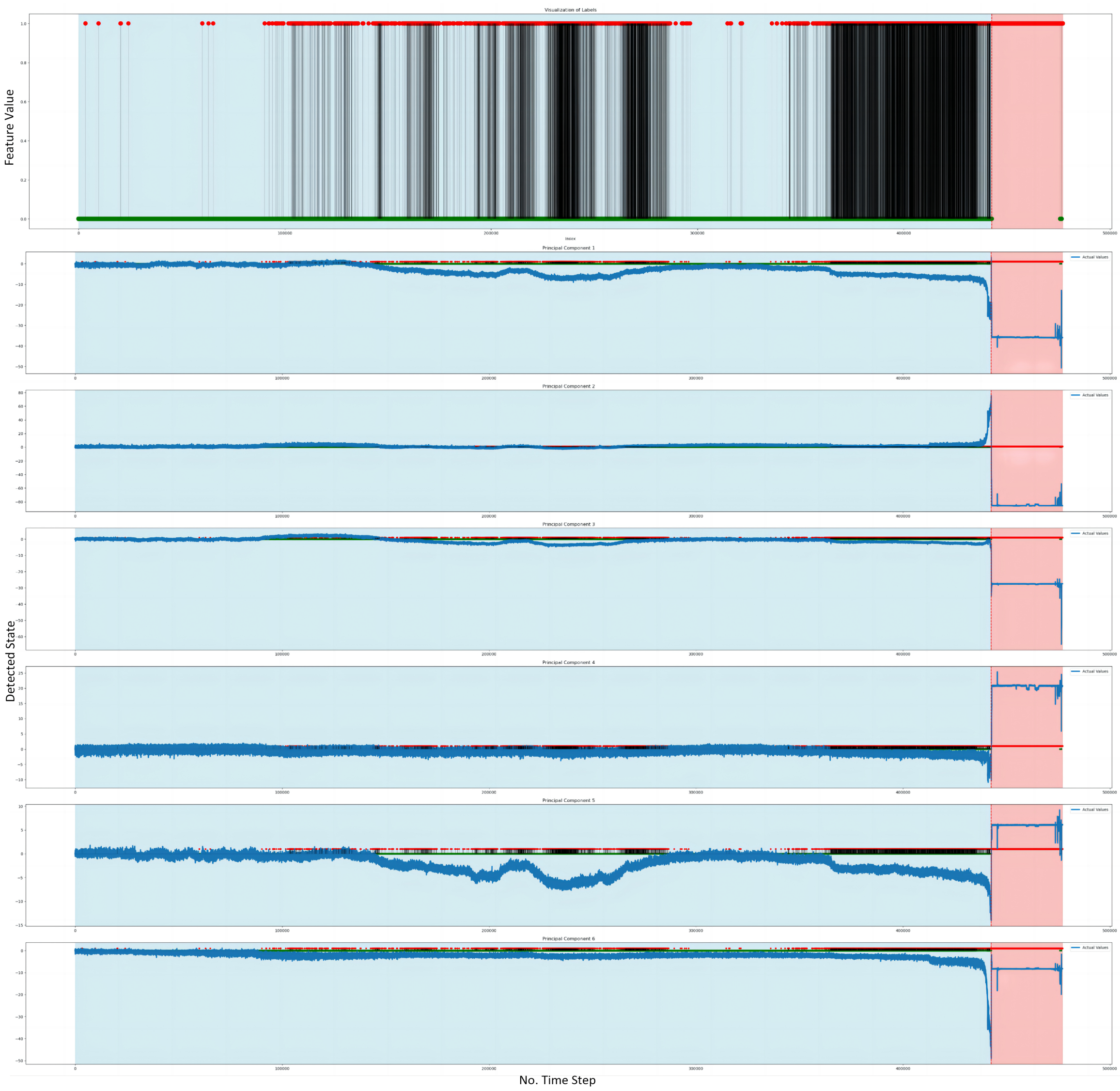

Figure G.3: CDMEVENT0062023: Short-term Detection Results for VARIMA + One-class SVM with PCA and New Feature Values

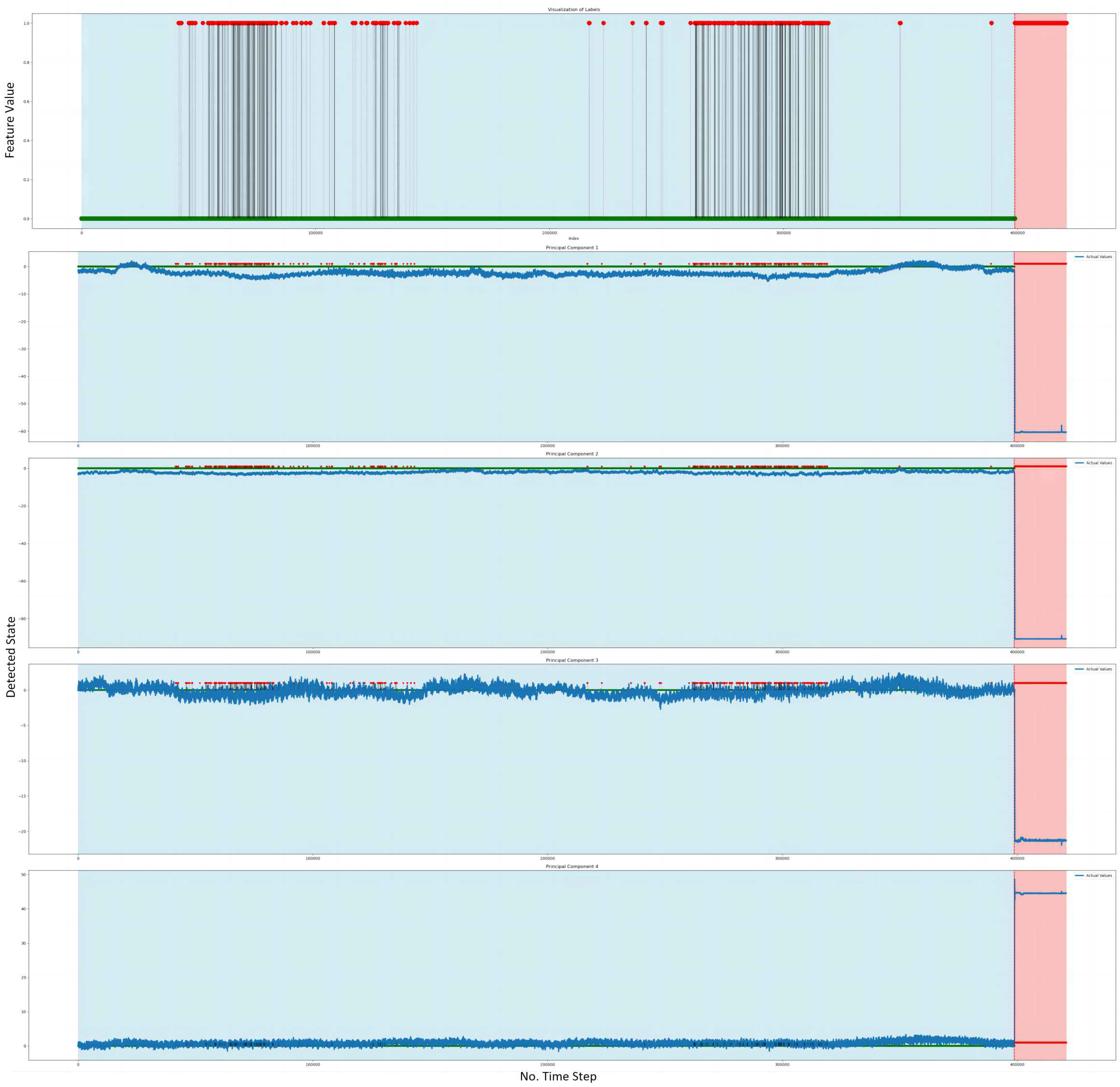

Figure G.4: CDSEVENT0052022: Short-term Detection Results for VARIMA + One-class SVM with PCA and New Feature Values

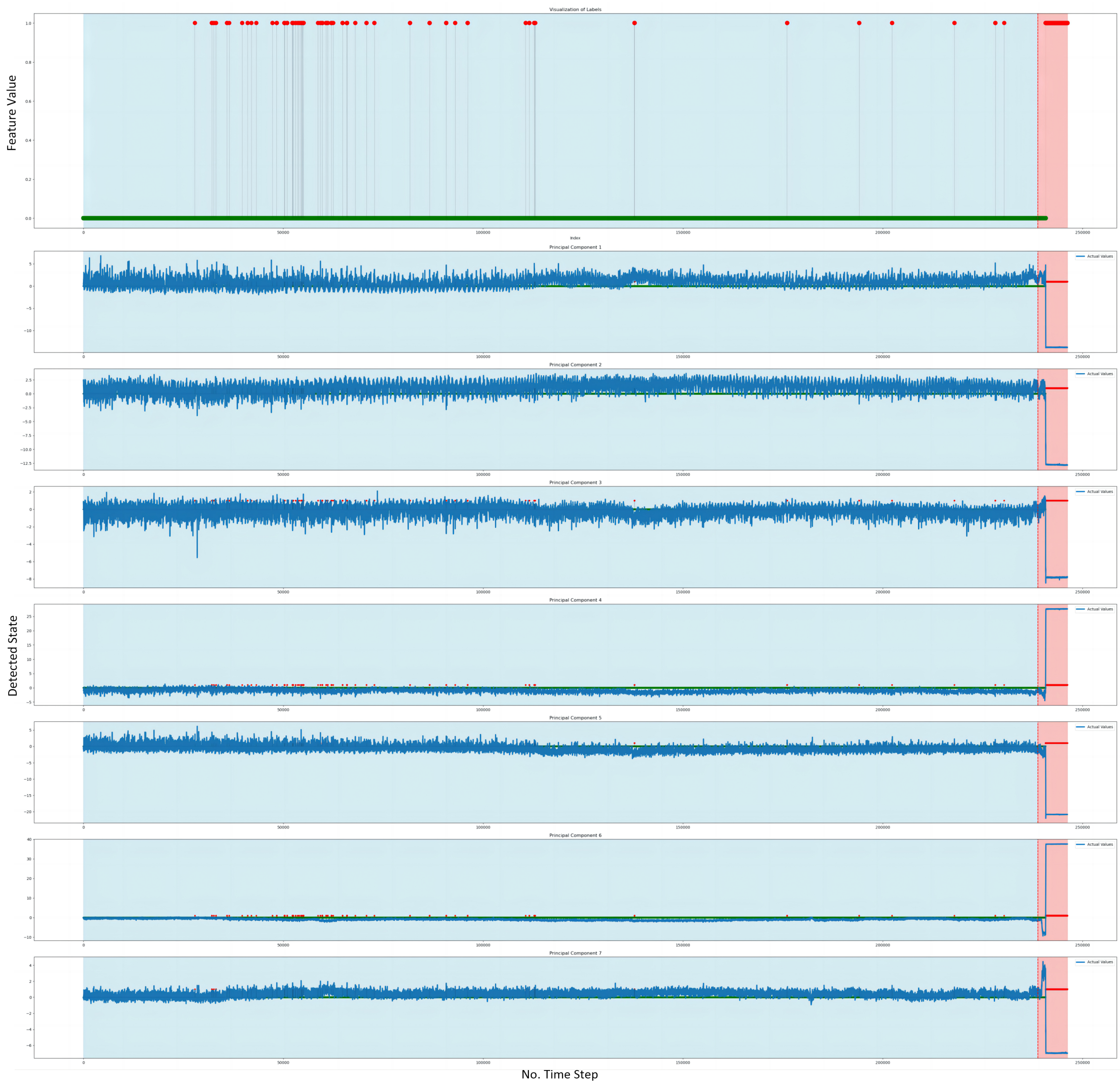

Figure G.5: CDSEVENT0072022: Short-term Detection Results for VARIMA + One-class SVM with PCA and New Feature Values

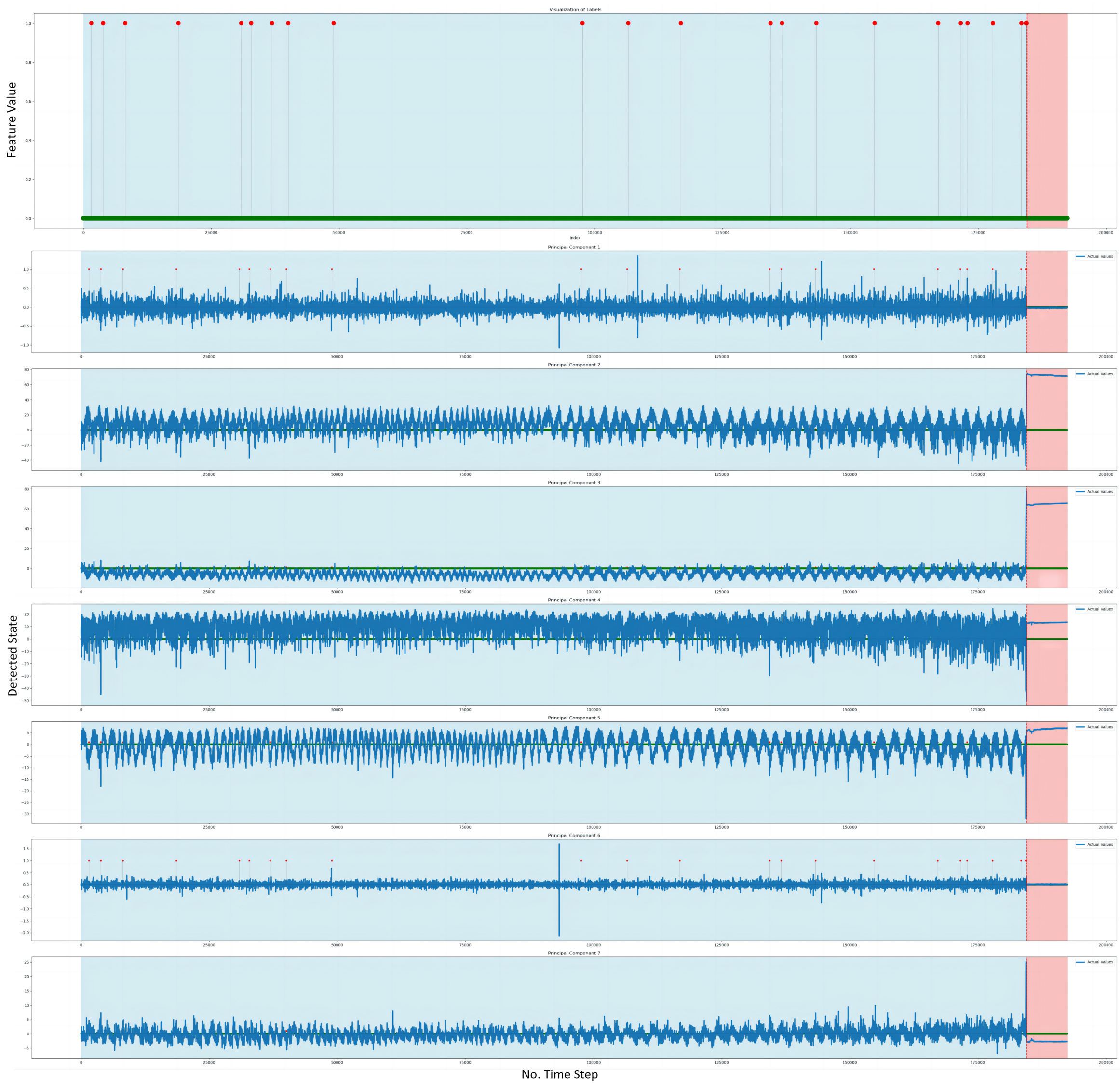

Figure G.6: CDSEVENT0102023: Short-term Detection Results for VARIMA + One-class SVM with PCA and New Feature Values

H

Short-term Detection Results: VARIMA + One-class SVM without PCA

Table H.1: Performance of VARIMA + One-class SVM without PCA for Different Datasets

Dataset	Accuracy	Precision	Recall	F1
CDIEVENT0012021	0.9979	0.9940	0.9645	0.9788
CDMEVENT0012022	0.9888	0.9878	0.9458	0.9656
CDMEVENT0062023	0.9892	0.9364	0.9927	0.9624
CDSEVENT0052022	0.9974	0.9791	0.9958	0.9873
CDSEVENT0072022	0.9997	0.9945	0.9998	0.9972
CDSEVENT0102023	0.9598	0.9525	0.5109	0.5111
	Model Training/s	Abnormality Detection/s		
CDIEVENT0012021	1146.4316	422.7645		
CDMEVENT0012022	565.0197	113.0982		
CDMEVENT0062023	1370.2862	672.0620		
CDSEVENT0052022	941.4703	497.6890		
CDSEVENT0072022	839.2274	738.4196		
CDSEVENT0102023	722.9331	148.2463		
	Detection Steps	Detection per Time Step		
CDIEVENT0012021	363007	0.001164618		
CDMEVENT0012022	226898	0.000498454		
CDMEVENT0062023	500409	0.001343025		
CDSEVENT0052022	441913	0.001126215		
CDSEVENT0072022	258727	0.002854049		
CDSEVENT0102023	202179	0.000733243		

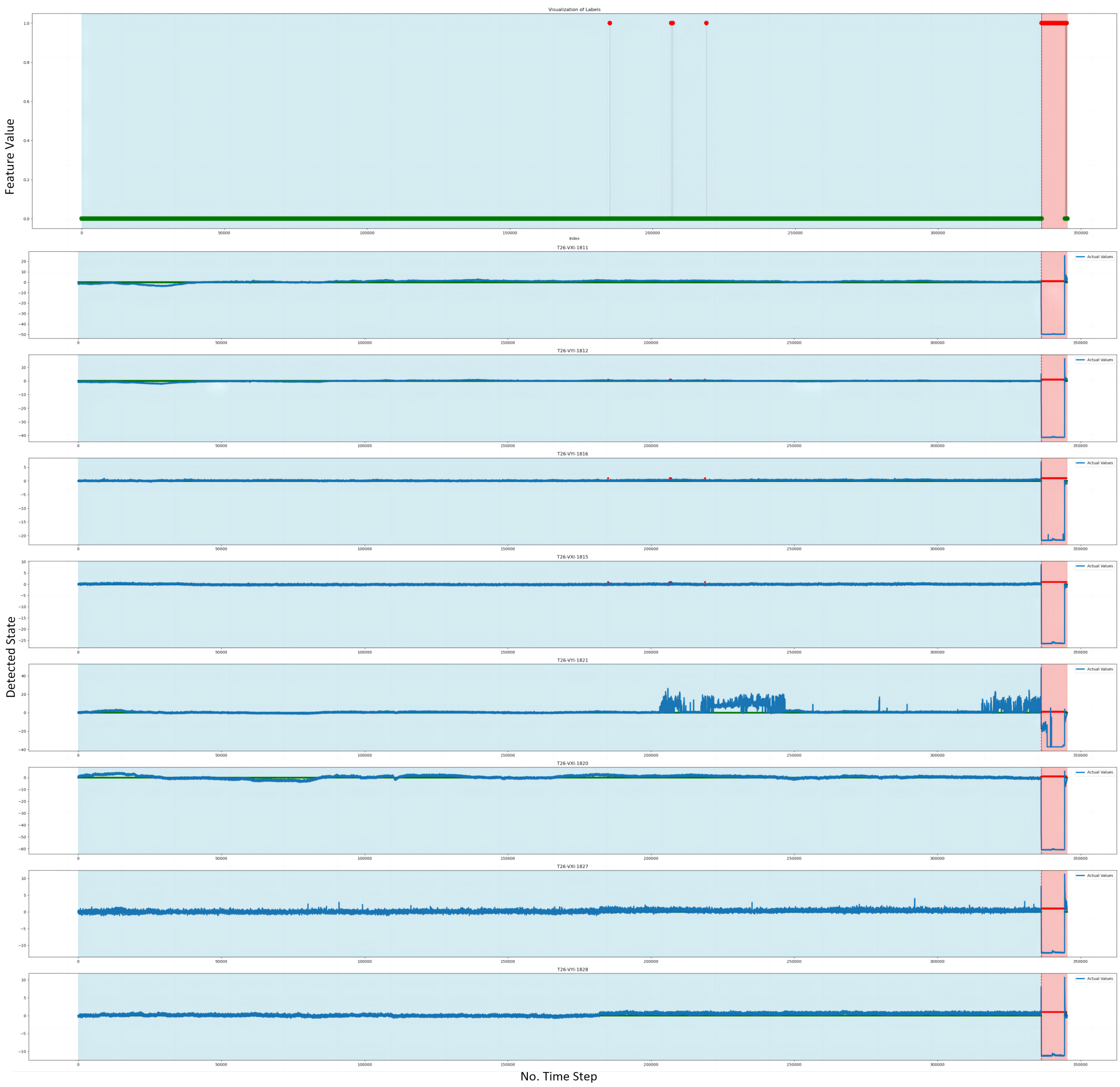

Figure H.1: CDIEVENT0012021: Short-term Detection Results for VARIMA + One-class SVM without PCA and New Feature Values

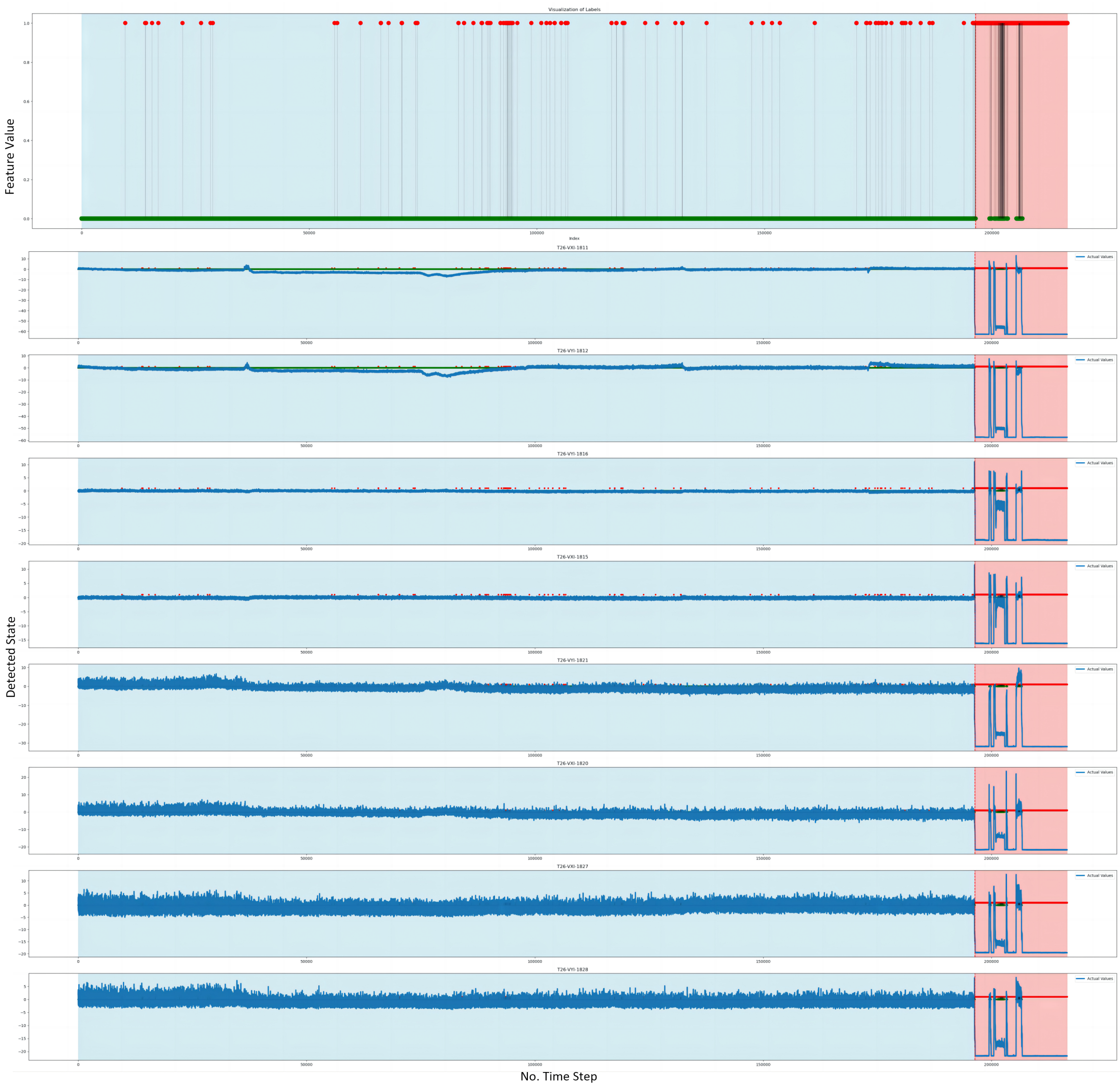

Figure H.2: CDMEVENT0012022: Short-term Detection Results for VARIMA + One-class SVM without PCA and New Feature Values

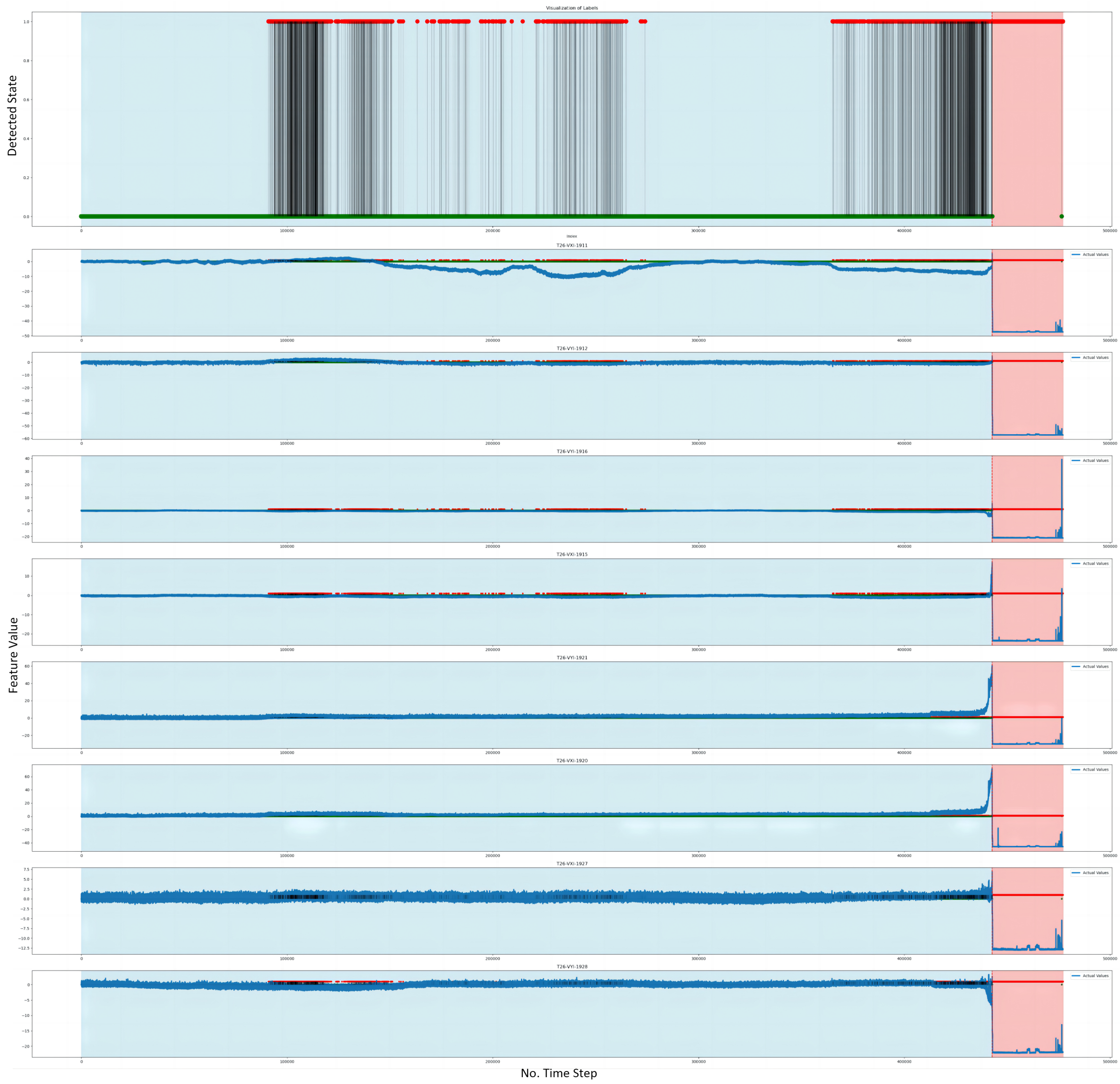

Figure H.3: CDMEVENT0062023: Short-term Detection Results for VARIMA + One-class SVM without PCA and New Feature Values

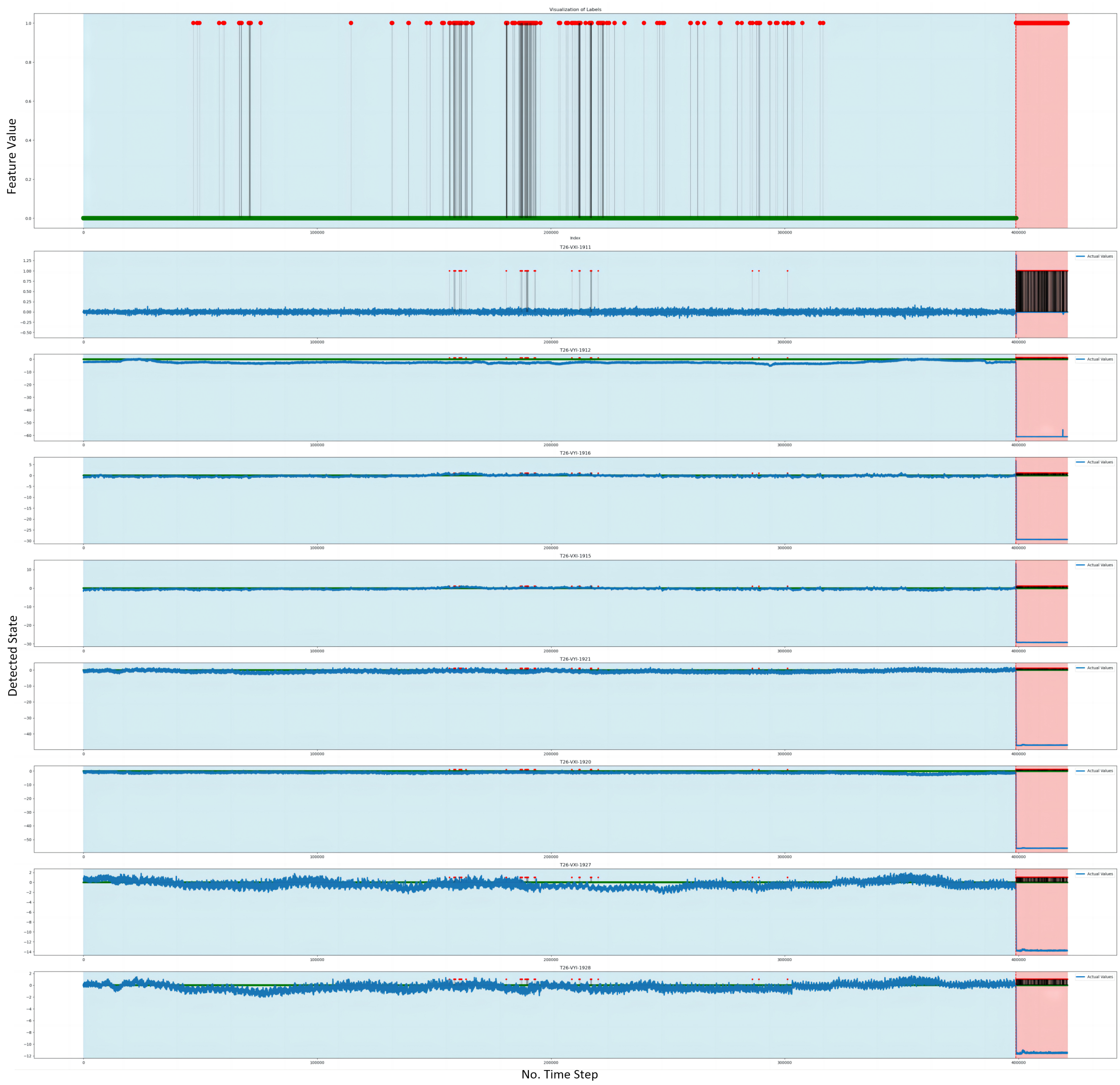

Figure H.4: CDSEVENT0052022: Short-term Detection Results for VARIMA + One-class SVM without PCA and New Feature Values

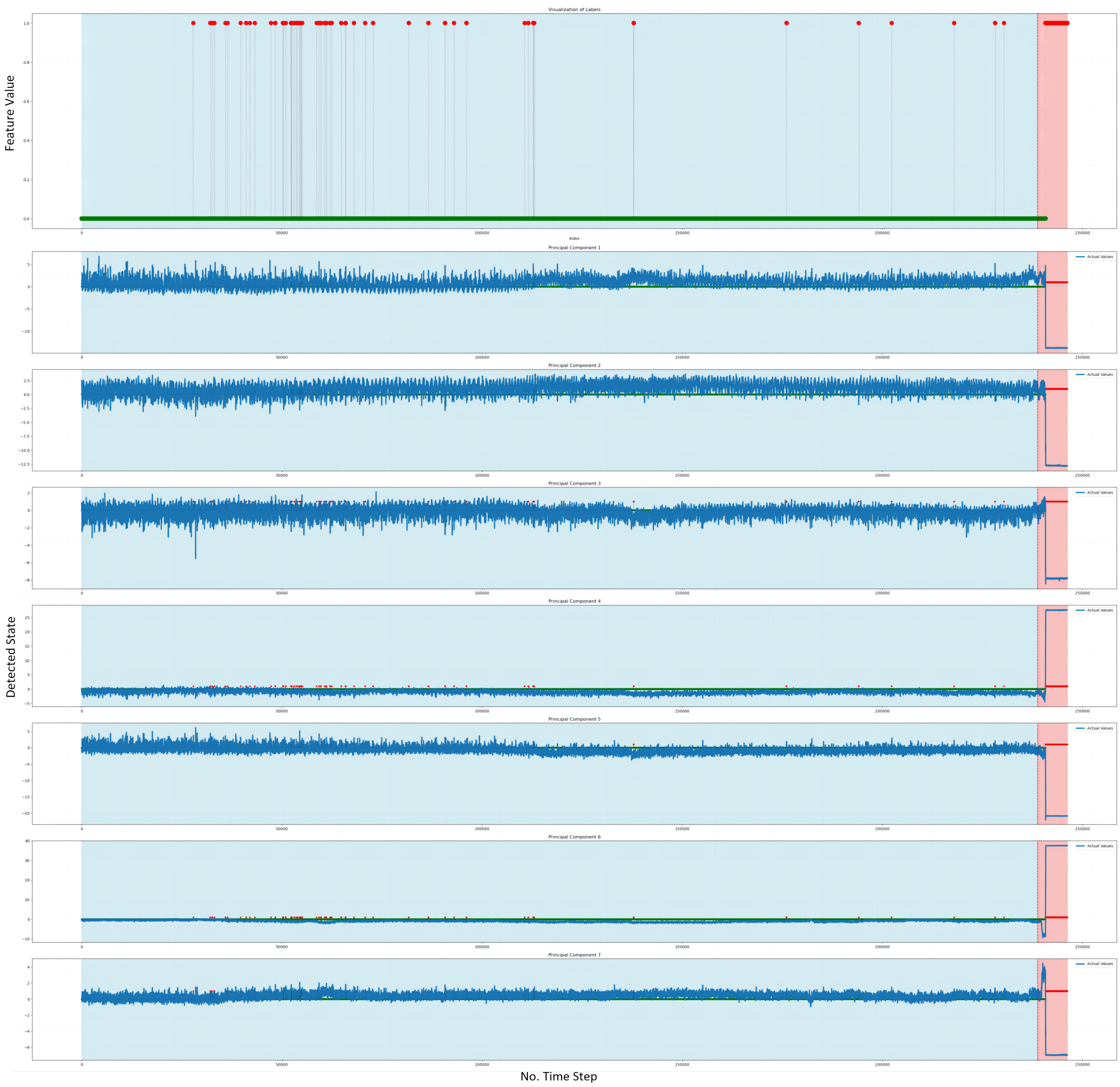

Figure H.5: CDSEVENT0072022: Short-term Detection Results for VARIMA + One-class SVM without PCA and New Feature Values

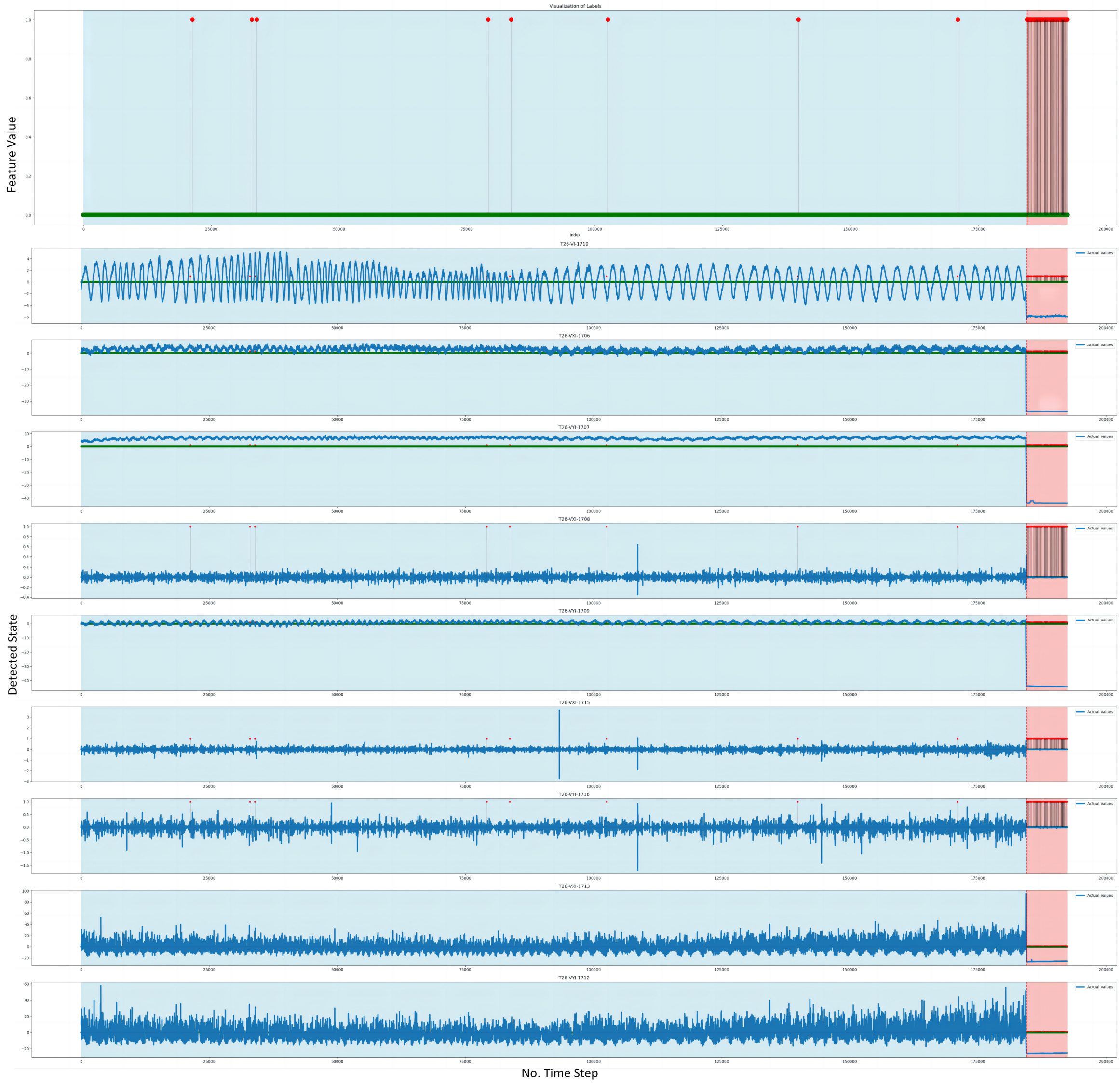

Figure H.6: CDSEVENT0102023: Short-term Detection Results for VARIMA + One-class SVM without PCA and New Feature Values

I

Short-term Detection Results: LSTM + One-class SVM with PCA

Table I.1: Performance of LSTM + One-class SVM with PCA for Different Datasets

Dataset	Accuracy	Precision	Recall	F1
CDIEVENT0012021	0.8689	0.5690	0.8396	0.5862
CDMEVENT0012022	0.9782	0.9557	0.9120	0.9325
CDMEVENT0062023	0.9823	0.8751	0.9880	0.9230
CDSEVENT0052022	0.9897	0.9301	0.9411	0.9315
CDSEVENT0072022	0.9986	0.9712	0.9993	0.9848
CDSEVENT0102023	0.9472	0.9215	0.5020	0.4906
	Model Training/s	Abnormality Detection/s		
CDIEVENT0012021	7584.3937	347.7925		
CDMEVENT0012022	1383.5731	119.2618		
CDMEVENT0062023	9716.3895	370.3922		
CDSEVENT0052022	2716.3895	172.3632		
CDSEVENT0072022	3830.3350	234.2928		
CDSEVENT0102023	3130.3936	156.1959		
	Detection Steps	Detection per Time Step		
CDIEVENT0012021	363007	0.000958088		
CDMEVENT0012022	226898	0.000525619		
CDMEVENT0062023	441915	0.000838153		
CDSEVENT0052022	500409	0.000344445		
CDSEVENT0072022	258726	0.000905563		
CDSEVENT0102023	202238	0.000772337		

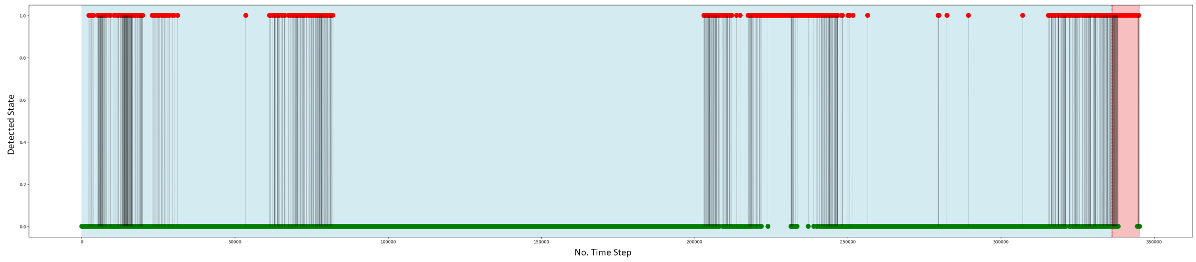

Figure I.1: CDIEVENT0012021: Short-term Detection Results for LSTM + One-class SVM with PCA

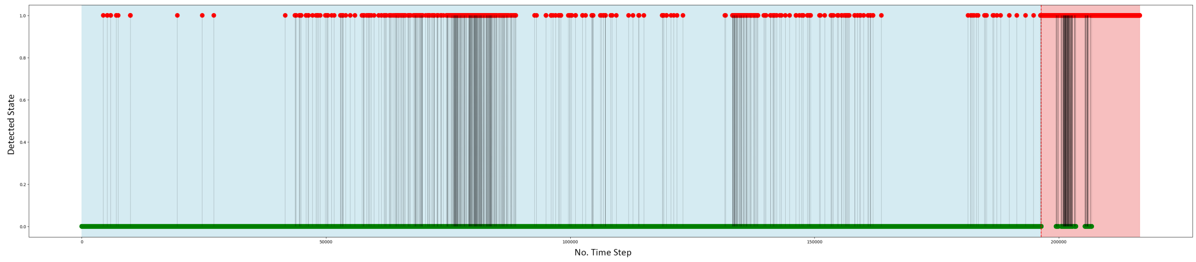

Figure I.2: CDMEVENT0012022: Short-term Detection Results for LSTM + One-class SVM with PCA

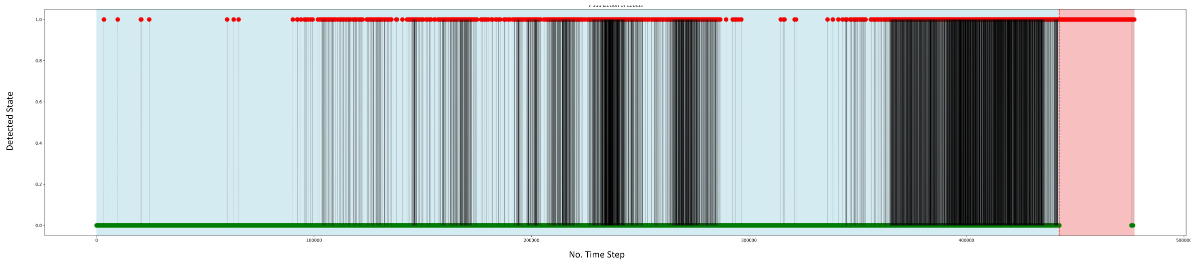

Figure I.3: CDMEVENT0062023: Short-term Detection Results for LSTM + One-class SVM with PCA

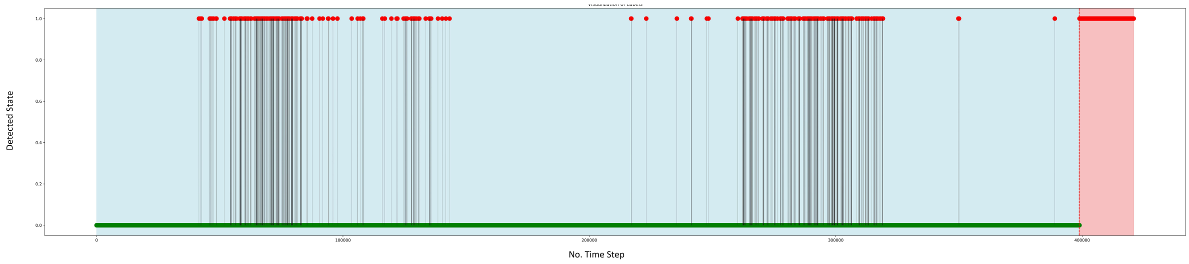

Figure I.4: CDSEVENT0052022: Short-term Detection Results for LSTM + One-class SVM with PCA

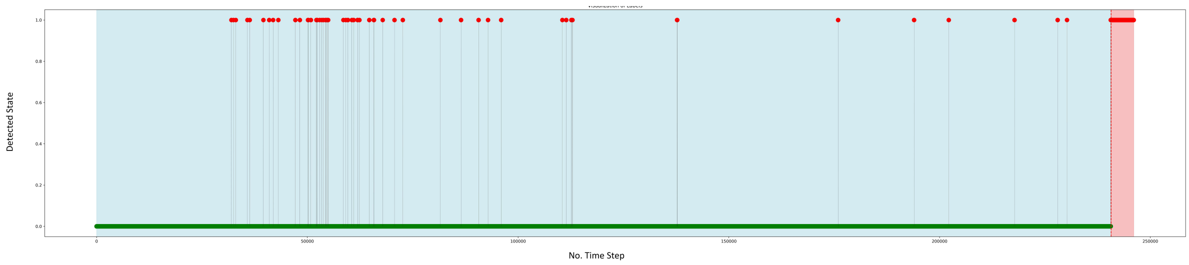

Figure I.5: CDSEVENT0072022: Short-term Detection Results for LSTM + One-class SVM with PCA

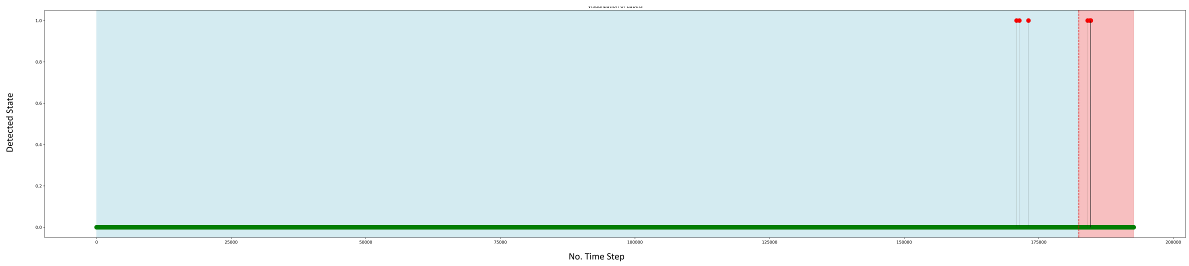

Figure I.6: CDSEVENT0102023: Short-term Detection Results for LSTM + One-class SVM with PCA

J

Short-term Detection Results: LSTM + One-class SVM without PCA

Table J.1: Performance of LSTM + One-class SVM without PCA for Different Datasets

Dataset	Accuracy	Precision	Recall	F1
CDIEVENT0012021	0.9979	0.9951	0.9647	0.9794
CDMEVENT0012022	0.989	0.9885	0.9459	0.966
CDMEVENT0062023	0.9986	0.9897	0.9965	0.993
CDSEVENT0052022	0.9902	0.9801	0.9897	0.9831
CDSEVENT0072022	0.9995	0.9907	0.9997	0.9952
CDSEVENT0102023	0.9994	0.9938	0.9997	0.9967
	Model Training/s	Abnormality Detection/s		
CDIEVENT0012021	8567.6785	611.9176		
CDMEVENT0012022	1944.2742	192.4267		
CDMEVENT0062023	1654.647	818.0739		
CDSEVENT0052022	1043.3751	524.6771		
CDSEVENT0072022	4118.9375	288.0048		
CDSEVENT0102023	2074.1251	256.5581		
	Detection Steps	Detection per Time Step		
CDIEVENT0012021	363007	0.001685691		
CDMEVENT0012022	226898	0.000848076		
CDMEVENT0062023	441915	0.001851202		
CDSEVENT0052022	500409	0.0010485		
CDSEVENT0072022	258726	0.001113165		
CDSEVENT0102023	202181	0.001268953		

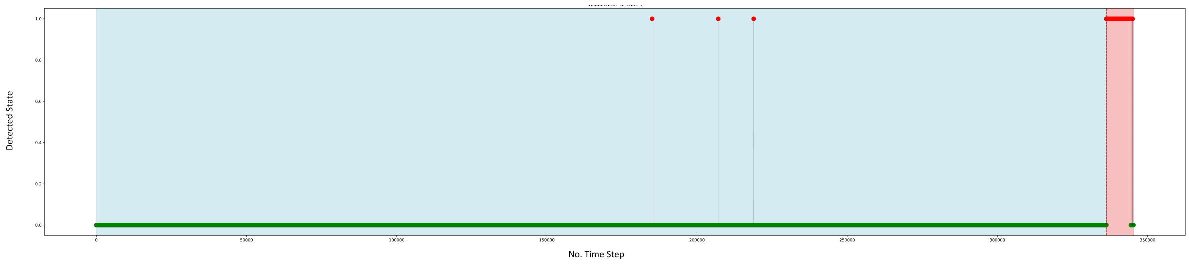

Figure J.1: CDIEVENT0012021: Short-term Detection Results for LSTM + One-class SVM without PCA

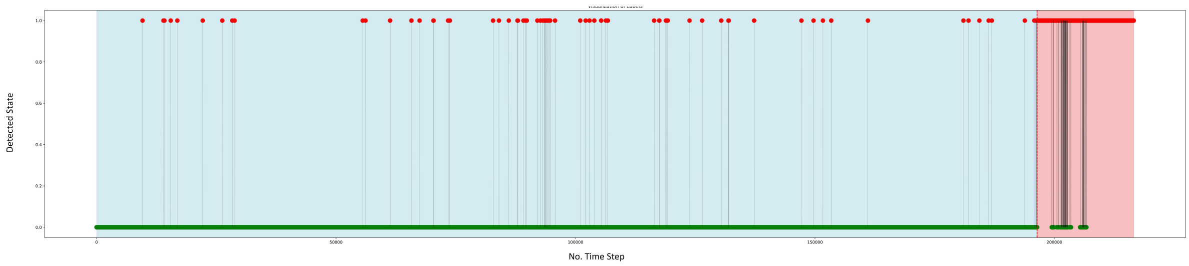

Figure J.2: CDMEVENT0012022: Short-term Detection Results for LSTM + One-class SVM without PCA

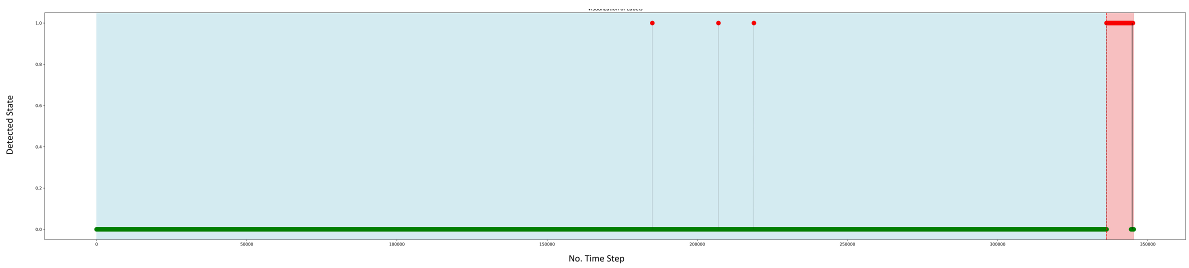

Figure J.3: CDMEVENT0062023: Short-term Detection Results for LSTM + One-class SVM without PCA

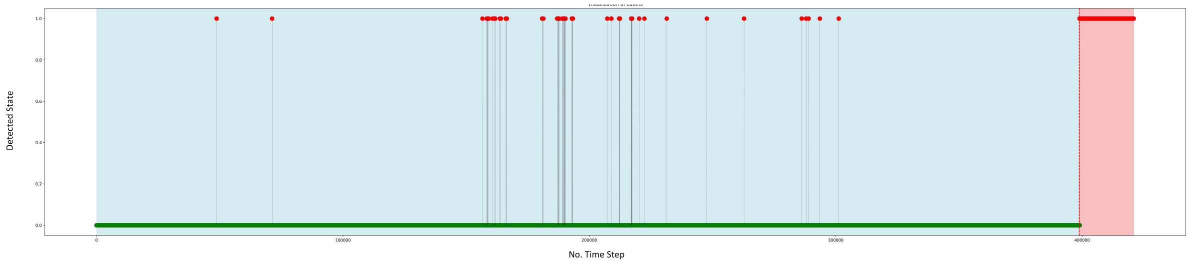

Figure J.4: CDSEVENT0052022: Short-term Detection Results for LSTM + One-class SVM without PCA

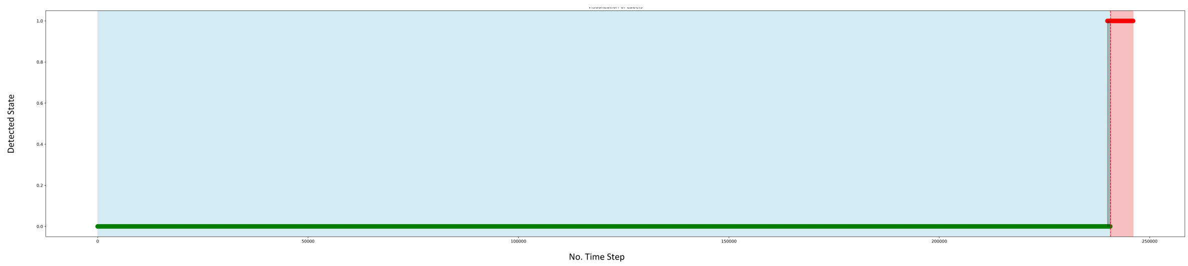

Figure J.5: CDSEVENT0072022: Short-term Detection Results for LSTM + One-class SVM without PCA

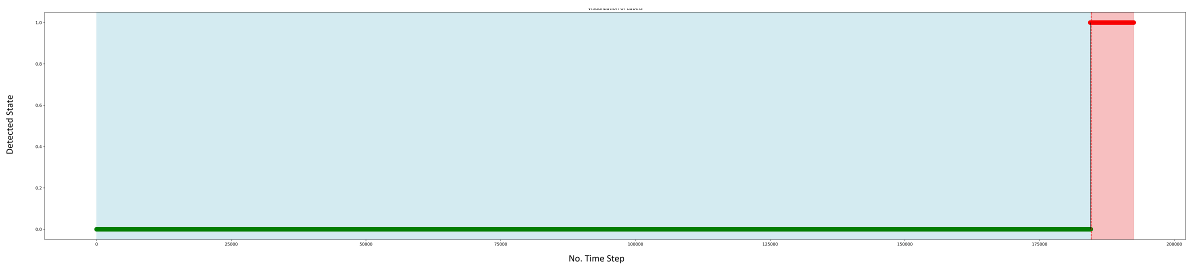

Figure J.6: CDSEVENT0102023: Short-term Detection Results for LSTM + One-class SVM without PCA

K

Flowcharts for Evaluation

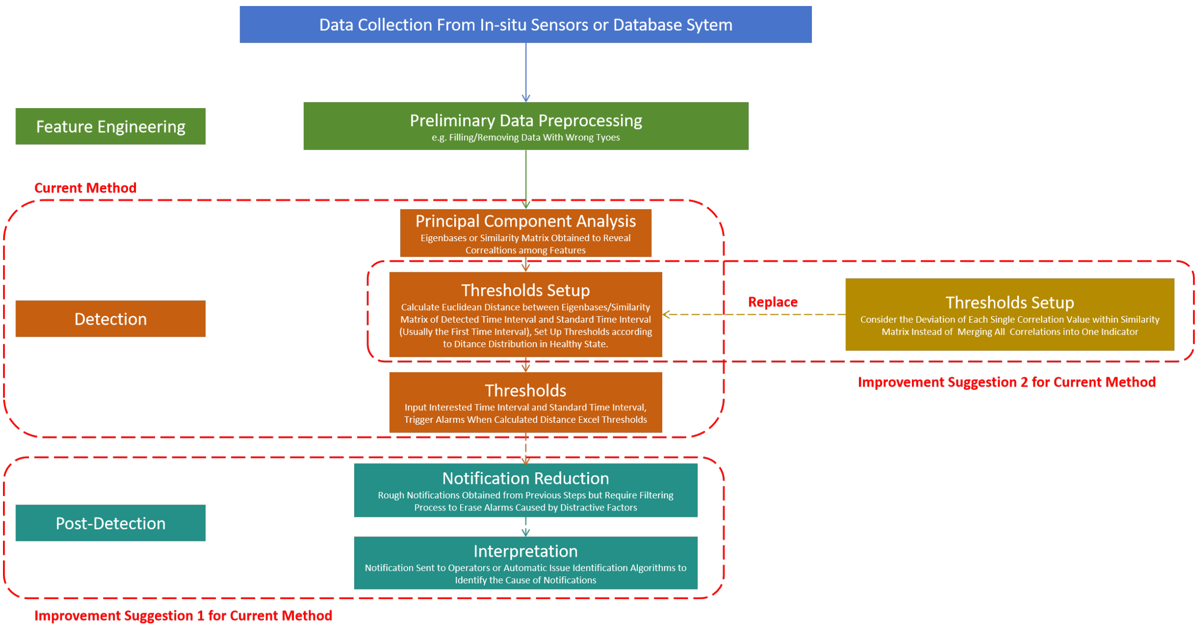

Figure K.1: Flowchart of PCA-based Prognosis, Including Current Method and Improvement Suggestions. Enhancement from Detection to Prediction not Mentioned Due to Insufficient Detective Performance.

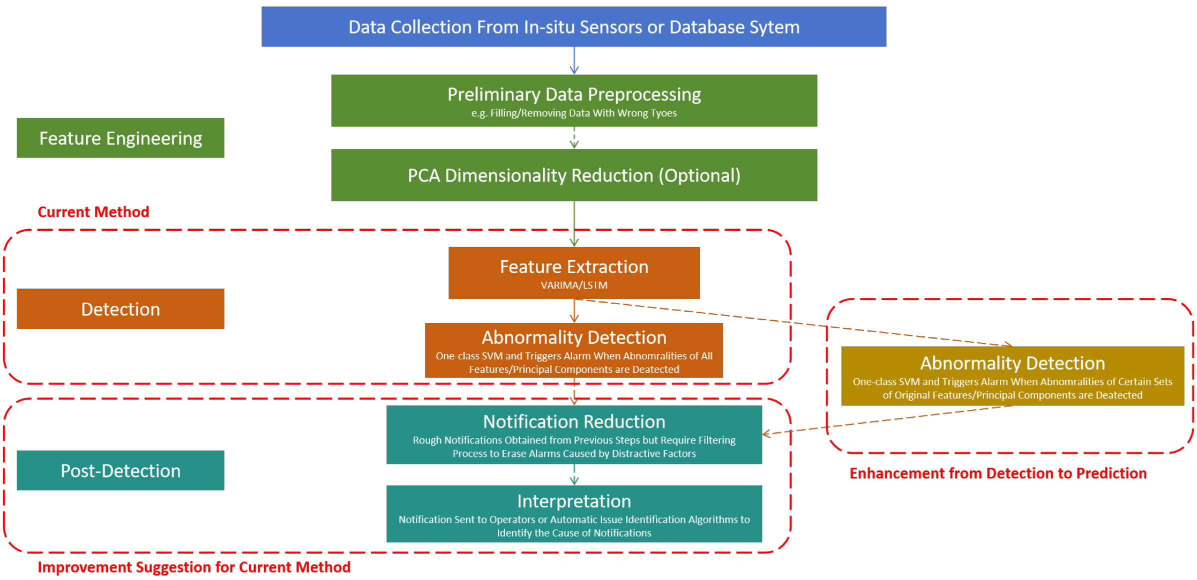

Figure K.2: Flowchart of VARIMA/LSTM + One-class SVM, Including Current Method, Improvement Suggestions and Enhancement.