



CHALMERS
UNIVERSITY OF TECHNOLOGY



Assessment of a Hybrid HMM-GMM and LSTM Pipeline for State Identification and Predictive Maintenance in Shaft Alignment

A case study in predictive maintenance for shaft alignment at Easy-Laser AB

Master's thesis in System, control and mechatronics

Tom Önnermalm & Simon Eriksson

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2026

www.chalmers.se

MASTER'S THESIS 2026

Assessment of a Hybrid HMM-GMM and LSTM Pipeline for State Identification and Predictive Maintenance in Shaft Alignment

A case study in predictive maintenance for shaft alignment at
Easy-Laser AB

Tom Önnermalm
Simon Eriksson



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical engineering
Division of Systems and control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Assessment of a Hybrid HMM-GMM and LSTM Pipeline for State Identification
and Predictive Maintenance in Shaft Alignment

A case study in predictive maintenance for shaft alignment at Easy-Laser AB

TOM ÖNNERMALM

SIMON ERIKSSON

© TOM ÖNNERMALM, 2026.

© SIMON ERIKSSON, 2026.

Supervisor: Per-Ove Ringsby, Easy-Laser AB

Examiner: Petter Falkman, Chalmers Department of Electrical Engineering

Master's Thesis 2026

Department of Electrical Engineering

Division of Systems and Control

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Cover: Easy Laser XT770 unit mounted on motor and pump setup.

Typeset in L^AT_EX

Printed by Chalmers Reproservice

Gothenburg, Sweden 2026

Abstract

This thesis provides an implementation and evaluation of a predictive pipeline consisting of Long Short-Term Memory (LSTM) networks and Hidden Markov Model with Gaussian emissions (HMM-GMM) for predicting anomalies and operating states in shaft alignment. The study focuses on a motor-pump setup, utilizing the Easy-Laser XT770 laser measurement platform for data acquisition as well as external temperature readings. Monitoring shaft alignment is critical, as misalignment leads to increased energy consumption and accelerated wear on bearings and other mechanical components. Empirical data collected from a physical setup provided initial insights into real-world behavior of the system. However, the algorithms were development centered around simulated data. Consequently, a simulation environment was designed to emulate a variety of physical systems, providing diverse datasets for training and evaluation.

In the proposed architecture, the LSTM model is employed for forecasting future alignment trends, while the HMM-GMM is utilized for state identification and anomaly detection. A marginalized variant of the HMM-GMM was also developed to perform these tasks without the continuous requirement of the laser measurement units. While the models yielded promising results in simulated environments, performance deviated when applied to empirical data, likely due to a model mismatch between the simulation and the physical system.

The results suggest that the HMM-GMM can function as a preliminary framework for identifying states and anomalies. Although further refinement is needed before practical deployment, the study demonstrates the potential of using a single measurement session of healthy states to initialize the model for real-world applications.

Keywords: Predictive Maintenance, Machine Learning, Shaft alignment, Hidden Markov model, Gaussian Mixture Model, LSTM

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
BIC	Bayesian Information Criterion
BLE	Bluetooth Low Energy
EM	Expectation-Maximization
HMM-GMM	Hidden Markov model with Gaussian emissions
IR	infrared
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Square Error
PdM	Predictive Maintenance
PSD	Position sensitive detectors
RNN	Recurrent Neural Networks
RPM	Revolutions Per Minute

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

m, p	Motor and pump notation
x, y	PSD x and y coordinate

Sets

$\mathcal{D}_{real}$	Set of real processed data
$\mathcal{D}_{train}, \mathcal{D}_{val}, \mathcal{D}_{test}$	Sets of simulated data

Variables

A_{jump}	Magnitude of initial movement in PSDs
B_m, B_p	Bias in motor and pump PSD sensor readings
C	Coupling effect between motor and pump
d_m, d_p	Magnitude readings from motor and pump PSDs
$\hat{d}_m, \hat{d}_p$	Predicted magnitude readings for motor and pump PSDs
G_m, G_p	Tilt amplifier for motor and pump PSD simulated readings
h	Length of prediction horizon
k_m	Thermal coefficients for motor
k_p	Thermal coefficients for pump
l	Length of known sequence
L	Distance between PSD measuring units
N	Number of simulated sequences in dataset

P_m, P_p	Motor and pump position
R_m, R_p	Rotation matrix for motor and pump
S_m, S_p	Readings from motor and pump PSDs
t	Index for time step
$t_{sampling}$	Resampling window
T_a	Ambient temperature
T_m, T_p	Motor and pump temperature
$\hat{T}_m, \hat{T}_p$	Predicted motor and pump temperature
$T_{m,max}, T_{p,max}$	Motor and pump maximum temperature
$T_{m,start}, T_{p,start}$	Motor and pump initial temperature
$T_{m,target}, T_{p,target}$	Motor and pump target temperature
$v_{m,exp}, v_{p,exp}$	Thermal expansion for motor and pump
V_m, V_p	Velocity readings from motor and pump PSDs
W	Position noise
$x_{m,eq}, x_{p,eq}$	Motor and pump expansion equilibrium
α_m, β_m	Motor expansion parameters
α_p, β_p	Pump expansion parameters
Δt	Sampling frequency in simulation
$\Delta T_m, \Delta T_p$	Motor and pump temperature derivatives
ϵ	Time window for neighbor pairing
$\theta_{m,tilt}, \theta_{p,tilt}$	Tilt angle of motor and pump PSD w.r.t. gravitational axis
ζ_m, ζ_p	Thermal time constants for motor and pump
ω	Axis rotational speed (RPM)

Contents

List of Acronyms	vii
Nomenclature	ix
List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Background	1
1.2 Aim	2
1.3 Research Questions	2
1.4 Limitations, Assumptions and Boundaries	3
1.5 Ethical Aspects	3
1.6 Environmental Aspects	4
2 Experimental Setup and Data Acquisition	5
2.1 System Description	5
2.2 Measurement System	6
2.3 Dataset Acquisition	7
2.4 Dataset Pre-processing	7
3 Simulation Model	11
3.1 Mathematical Model	11
3.1.1 Temperature Dynamics	12
3.1.2 Heat Bias and Equilibrium Points	13
3.1.3 Machine Dynamics	13
3.1.4 PSD Readings	14
3.2 Data Generated from Simulation Model	15
3.2.1 PSD Simulation	15
3.2.2 Temperature Simulation	16
3.2.3 Standardization Methods	17
4 Algorithm Design	19
4.1 Alignment Prediction	19
4.1.1 The LSTM Model	20
4.1.2 LSTM Prediction Model Implementation	21

4.1.2.1	Tuning Process	22
4.2	State Identification and Anomaly Detection	24
4.2.1	Hidden Markov Model for State Transitions	25
4.2.2	Gaussian Mixture Emissions	26
4.2.3	Model Implementation	27
4.2.4	Model Evaluation	27
4.3	Pipeline	29
4.3.1	Testing of Pipeline	29
5	Results	31
5.1	LSTM Performance and Prediction Accuracy	31
5.1.1	LSTM Tuning	31
5.1.2	Performance of tuned model	36
5.2	HMM-GMM Evaluation and State Identification	38
5.2.1	Baseline Model Stability	38
5.2.2	State Construction and Reconstruction Ability	39
5.2.3	Feature Set Comparison	40
5.2.4	Anomaly Detection	43
5.3	Pipeline Evaluation	45
6	Discussion	49
6.1	LSTM as a Method for Prediction of Alignment	49
6.2	HMM-GMM as a Method for State Identification and Anomaly De- tection.	50
6.3	Comparative Analysis of Pipeline Performance	51
6.4	Research Questions	52
6.5	Future Work	53
	References	55

List of Figures

1.1	Easy Laser XT770 measurement unit kit.	1
2.1	Easy Laser XT770 unit mounted on motor-pump setup.	5
2.2	Illustration of misalignment types in laser measurement system. . . .	6
2.3	Comparison of filtered motor and pump PSD data using different resampling durations ($t_{sampling} \in \{1, 10, 30\}$). Increased sampling lengths demonstrate a reduction in measurement noise.	8
3.1	Comparison of simulated vs real data.	15
3.2	Different example PSD trajectories from the simulated machine setup.	16
3.3	Different examples of temperature and RPM characteristics from simulations.	16
4.1	Illustration of information pipeline.	19
4.2	Illustration of LSTM input and output information. X denotes the input feature set for the LSTM, while Y represents the predicted output. l represents the length of the known sequence and h represents the length of the prediction horizon.	20
4.3	Diagram of LSTM architecture [17].	21
5.1	LSTM model prediction comparison based on input case, for sequence 1 from $\mathcal{D}_{test}$	33
5.2	LSTM model prediction comparison based on input case, for sequence 2 from $\mathcal{D}_{test}$	33
5.3	LSTM model prediction comparison based on input case, for sequence 3 from $\mathcal{D}_{test}$	34
5.4	Final model estimation on sequence form $\mathcal{D}_{test}$	36
5.5	Final model estimation on sequence form $\mathcal{D}_{test}$	37
5.6	Final model estimation on sequence form $\mathcal{D}_{test}$	37
5.7	Simulated results showing temperature and alignment behavior. . . .	38
5.8	State transitions when model training is conducted on different data lengths.	40
5.9	States presented in the temperature data 4 states, 3 mixtures. . . .	41
5.10	States presented in the temperature 7 states, 2 mixtures.	41
5.11	Simulated results showing temperature and PSD behavior of simulation with added abnormal behavior.	44
5.12	Data from two different measuring sessions (Blue: $\mathcal{D}_{real_A}$, Red: $\mathcal{D}_{real_B}$). .	45

5.13	Pipeline evaluation on simulated data. The upper plot displays PSD magnitudes for motor and pump, while the lower plot shows temperature development. Background colors indicate the HMM-GMM identified states, where during training 2 states and 1 mixture was identified. Without anomaly inclusion.	46
5.14	Full pipeline using full HMM-GMM model, with anomalies introduced.	46
5.15	Full pipeline using marginalized HMM-GMM model, with anomalies introduced.	47
5.16	Pipeline evaluation on empirical data. Background colors indicate the HMM-identified states, where during training 4 states and 3 mixture was identified.	47

List of Tables

3.1	Comparison between scaled data metrics, utilizing different scalers fit to 1000 training sequences.	17
4.1	LSTM output definition.	22
4.2	LSTM input case definitions for different feature sets, with l representing the known sequence length.	23
4.3	LSTM dimensions.	24
4.4	Set size definitions.	24
4.5	Feature set example.	28
5.1	Baseline LSTM model configuration.	32
5.2	Experimental Results of MAE and MSE by Case and Loss Function.	34
5.3	Performance for input features Case 5 and 6, minmax scalers and MAE loss function, for different sizes of $\mathcal{D}_{\text{train}}^N$ and $\mathcal{D}_{\text{val}}^N$, with $\mathcal{D}_{\text{test}}$ of size 100 sequences.	35
5.4	Experimental Results: Performance dependent on Hidden Size of LSTM model.	35
5.5	LSTM model configuration after tuning.	36
5.6	Summary of model performance for selected test configurations.	39
5.7	Confusion Matrix for internal state consistency (50 000 samples). Row represent the full model and the columns represent the marginalized model.	39
5.8	Consistency and performance metrics of the models across different feature sets (50 000 samples).	41
5.9	Real-Time operating state identification performance of the models across different feature sets (50,000 samples).	42
5.10	HMM-GMM results for real data. Match ratio indicates the consistency between the full and marginalized model on the same measurement session.	42
5.11	Real-Time operating state identification performance for empirical data across different feature sets.	43
5.12	Anomaly detection on simulated data. The baseline feature set is used.	44
5.13	HMM-GMM results for anomaly detection on empirical data. The feature set $\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$ is used.	45
5.14	HMM-GMM results for anomaly detection on empirical data. The feature set $\{S_m, S_p, \sqrt{T_m^2 + T_p^2}\}$ is used.	45

5.15 Pipeline results for anomaly detection for full and maginalized models.	47
--	----

1

Introduction

Predictive Maintenance (PdM) powered by Artificial Intelligence (AI) has revolutionized industrial asset management, offering a more sustainable and cost-efficient alternative to traditional reactive strategies [1]. By leveraging AI for complex data analysis and anomaly detection, industries can now preemptively address mechanical degradations before they escalate into costly failures. A critical factor in this maintenance ecosystem is shaft alignment, a primary determinant of machine longevity and energy efficiency [2]. While Easy-Laser provides industry-leading laser measurement systems for precision alignment, the integration of these data streams into predictive AI frameworks remains an evolving frontier. This thesis evaluates a novel application for laser measurement technology within the PdM landscape, aiming to transform dynamic alignment data into predictive insights.



Figure 1.1: Easy Laser XT770 measurement unit kit.

1.1 Background

Shaft alignment of rotating machinery has been a rapidly growing field in the last decades. It has been a fundamental pillar in improving operational reliability, the reduction of wear and energy consumption in machinery [2]. Identifying the conditions under which shaft alignment changes, and the nature of those changes, is therefore of significant interest.

The process of shaft alignment involves the alignment of the rotational centerline of coupled shafts, such as motors and pumps, gearboxes, fans, etc. Ensuring that the alignment is collinear is critical to extend the lifespan of machinery, as even small deviations in misalignment causes additional stresses and internal rubbing, leading

to additional wear of components e.g. bearings, etc [2].

Today, there are numerous ways to measure misalignment, one of the most prevalent is with the use of laser-based systems and positional sensitive devices (PSD). Easy Laser's XT770 series [3] utilizes this technology to find the angular and offset error, and provides the user with instructions to align the shaft. These systems can only be used on one machine at a time, and to reduce the need for multiple systems, there is an interest in using other sensors to identify misalignment.

The rise of PdM has also ignited the interest in development of digital twins of machinery, in practice developing complex mathematical models able to reflect real machine behavior [4], [5]. These models are then used for diagnosis and testing. With this in mind, monitoring and modeling shaft alignment of rotating machinery digitally is an area that could enhance machine monitoring and maintenance strategies.

In this study the identification of operating states (cooling, heating, stable operation), misalignment anomalies and future deviations in shaft alignment are delved into. Utilizing a motor-pump setup as the subject for modeling and development of data driven algorithms.

1.2 Aim

The primary objective of this thesis is to investigate methods for identifying operational states and predicting shaft alignment in rotating machinery. Specifically, the study evaluates the utility of Easy-Laser measurement units and external temperature sensors within a predictive maintenance framework.

To achieve this, the aim is to develop a modular data pipeline that serves as a unified framework for both state identification, anomaly detection and temporal forecasting.

Furthermore, a key objective is to assess the ability to identify operational states and anomalies in the absence of active laser measurement units.

1.3 Research Questions

To address the challenges of industrial state monitoring and predictive alignment, this thesis investigates the following research questions:

1. To what extent can the future alignment of rotating machinery be predicted utilizing external temperature reading independently and in combination with laser measurement units.
2. To what extent can operating states and anomalies of rotating machinery be identified based on external temperature reading independently and in combination with laser measurement units.

1.4 Limitations, Assumptions and Boundaries

This project is constrained to the use of XT770 laser measurement units (shown in 1.1) in combination with temperature measurements obtained from an infrared (IR) sensor. The study focuses on a specific setup consisting of an electric motor coupled to a warm water pump. In this configuration, the pump temperature is assumed to be primarily influenced by the temperature of the circulating fluid rather than by operational inputs such as load or rotational speed.

Furthermore, it is assumed that the shaft returns to its nominal alignment at the beginning of each measurement session, as the measurement system captures relative deviations in PSD values with respect to the initial aligned state. Consequently, the analysis is based on changes in alignment rather than absolute positioning. It is also assumed that shaft misalignment is predominantly driven by thermal effects.

The availability of real-world data is limited, and the study therefore relies primarily on simulated datasets to develop and evaluate the proposed methods. While this enables controlled experimentation, it may limit the direct generalizability of the results to real industrial environments.

To maintain a manageable experimental scope, the study prioritizes the evaluation of input variables, model structure, and loss functions. Other parameters, such as batch size and input sequence length, are kept constant to ensure a consistent basis for comparison across experiments.

1.5 Ethical Aspects

Integrating data-driven algorithms into industrial maintenance introduces several ethical challenges, particularly regarding the reliability of automated decision-making. A primary concern is algorithmic bias, where models may favor specific outcomes based on skewed or non-representative training data [6]. In the context of shaft alignment, such bias could lead to discriminatory diagnostics, which systematically misidentifying faults in certain machine classes while overlooking others. This undermines the reliability of the system and can lead to unnecessary industrial downtime or, conversely, equipment failure.

Furthermore, the transparency of the black box poses an ethical dilemma. If an AI-driven system provides recommendations of intervention, the lack of interpretability may impact technicians ability of validating the decision. This leads to the critical question of accountability [7]. When a predictive model fails to signal a critical misalignment, the responsibility for resulting safety hazards or environmental impacts must be clearly defined between the developers, and the end-users.

1.6 Environmental Aspects

Several factors drive the industrial adaption of PdM. Moving towards an proactive mindset that anticipates and aims to resolve potential errors before they escalation into functional failures [8]. PdM is a important concept in the industry as it aims to reduce the costs associated with unplanned maintenance of machinery. Not only is it beneficial for companies as an economic incentive to increase the lifetime of machinery, but likewise beneficial for the environment. Which is reinforced by the reduction of resource consumption connected to manufacturing new machinery in comparison to replacing or correcting individual components in the existing machinery.

Beyond operational efficiency, PdM offers significant environmental benefits by optimizing energy consumption. Mechanical misalignment is a primary driver of parasitic energy losses, as it increases internal friction and heat generation, particularly within bearings. Research indicates that replacing or properly maintaining a flawed bearing can reduce a machine's energy consumption by as much as 10–14% [9]. Furthermore, the use of laser-based alignment to predict and prevent component failure extends the service life of machinery, thereby reducing the environmental burden associated with the manufacturing and disposal of replacement parts.

2

Experimental Setup and Data Acquisition

This chapter provides an overview of the experimental setup, including the laser measurement units, system configuration, and the data acquisition procedure. It also describes the methods used for post-processing the collected data, including the synchronization of position-sensitive detector (PSD) measurements with temperature data, as well as the filtering techniques applied to reduce noise.

2.1 System Description

The experimental setup, illustrated in Figure 2.1, comprises an ABB cast iron motor coupled with a Sulzer pump. The system delivers water at 100°C to a boiler, operating at a constant speed of approximately 2671 RPM against a backpressure of 40–45 bar. To ensure continuous flow, the pump is equipped with an overflow valve.



Figure 2.1: Easy Laser XT770 unit mounted on motor-pump setup.

2.2 Measurement System

This thesis relies on measurements from laser-based measurement units and external temperature readings. The methodology used with these devices is described in this section.

Laser Measurement Units

Laser-based measurement systems are widely used in alignment and displacement monitoring due to their high precision. In this setup, laser beams are projected onto PSDs, where deviations in the measured position correspond to mechanical displacements in the system.

These displacements can originate from both parallel and angular misalignment [10], as illustrated in Fig. 2.2. In practice, distinguishing between these types depends on the specific application, as certain combinations of translations and rotations can be mathematically equivalent. While this distinction is typically important for determining how a machine should be adjusted during alignment, the focus of this thesis is solely on identifying the presence of displacement, rather than classifying its type.

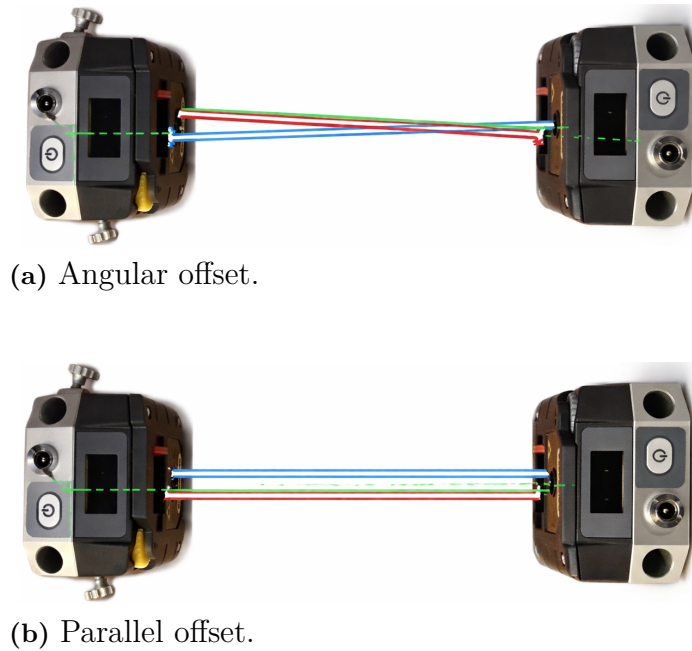


Figure 2.2: Illustration of misalignment types in laser measurement system.

The measurement units were mounted on the motor and pump using fixtures attached to flat surfaces via adhesive, as illustrated in Figure 2.1.

The measurement units can continuously stream data [3] to an external computer via Bluetooth Low Energy (BLE), where it is stored for subsequent analysis. The recorded data consists of the x and y coordinates of the laser spot on the PSDs, as

well as the roll angle of each measurement unit.

During measurements, the roll angles of both units are initialized to 0 degrees, and the laser beams are aligned to the center of the PSDs. This ensures that any subsequent deviations in the measured positions correspond directly to horizontal and vertical displacements in the system.

Temperature Measurements

The temperature was measured with a infrared (IR) thermometer. The readings was taken on the outside of the chassis on both the motor and the pump. Since the IR thermometer measures surface temperature, the recorded values represent the external temperature of the components rather than the internal operating temperature. While this introduces a thermal lag, since the thermal energy radiates through the chassis, the surface temperature were still considered a useful proxy for the thermal state of the system, as heat transfer through the chassis is expected to reflect overall operating conditions.

2.3 Dataset Acquisition

The dataset used in this study consists of measurements from the PSD detectors and temperature sensors. The PSD measurements were sampled at a rate of 12 Hz, providing continuous measurements of the laser spot position in the horizontal and vertical directions. In parallel, temperature measurements were recorded at one-minute intervals and stored for later analysis.

Several sampling session where conducted, where the machine setup was started from a cold state, to verify that the movement of the PSD values and system behavior was consistent. Because of system constraints, each data acquisition session was limited to a duration of 30 minutes.

2.4 Dataset Pre-processing

To create a dataset suitable for algorithmic development, the raw sensor data needed to be pre-processed. The methodology used is further presented in this section.

PSD Synchronization and Filtering

Since the laser measurement units operated independently without a common hardware trigger, the raw data streams were sampled asynchronously. This resulted in a non-aligned temporal distribution of (x,y) position coordinates for the PSDs. To facilitate analysis and development of algorithms, the datasets were post-processed using a simple temporal ordering algorithm, where a nearest-neighbor approach was taken [11]. Data points from the measurement units at the motor and pump were paired only when their respective timestamps fell within a predefined maximum

2. Experimental Setup and Data Acquisition

temporal threshold set to $\epsilon = 50ms$, effectively synchronizing the independent measurements into a unified time-series. A pseudo code implementation of the pairing algorithm is shown in Algorithm 1.

Algorithm 1 Synchronization Filter

```

def Pair Data ( $PSD_m, PSD_p, \epsilon$ ):
1:   Pairs = {empty}
2:   For  $S_m$  in  $PSD_m$ :
3:      $t_m = S_m.value('time')$ 
4:     Find  $S_p$  in  $PSD_p$  with timestamp  $t_p$  minimizing  $|t_m - t_p|$ 
5:     if  $|t_m - t_p| \leq \epsilon$ :
6:       pair = join( $S_m, S_p$ )
7:       Add pair to Pairs
8:     else:
9:       ignore  $S_m$ 
10:  Return Pairs

```

The paired measured PSD points are then subject to filtering by taking the average PSD value over a set value of samples. The data after filtering is described in Figure 2.3, where $t_{sampling} \in \{1, 10, 30\}$. This is done in order to get less noisy data.

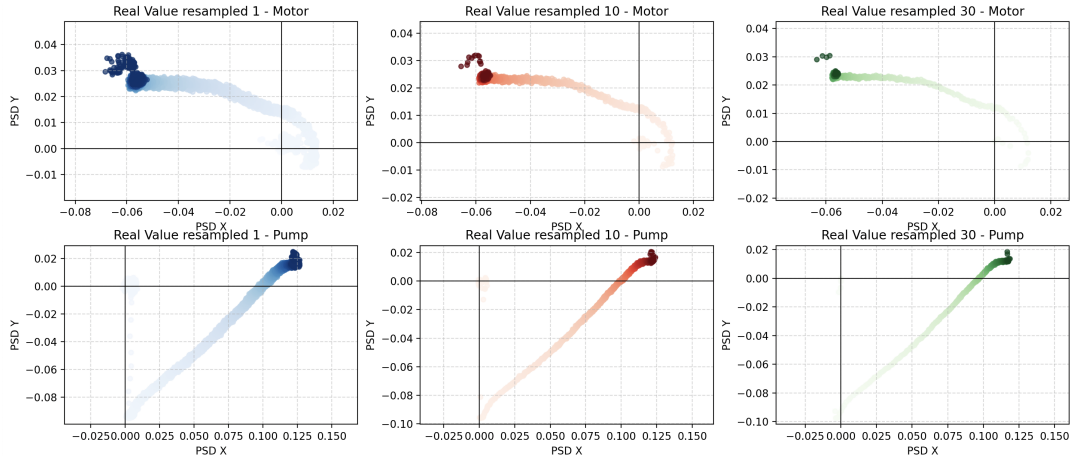


Figure 2.3: Comparison of filtered motor and pump PSD data using different resampling durations ($t_{sampling} \in \{1, 10, 30\}$). Increased sampling lengths demonstrate a reduction in measurement noise.

Additionally, the expansion velocity is added to each paring by $V_i(t) = \frac{S_i(t) - S_i(t-1)}{t_{sampling}}$, where $i \in [m, p]$.

Temperature Sampling

The sampling time of the temperatures of the motor and pump, T_m, T_p , was measured once per minute, in order to up sample this to a sampling frequency of once every $t_{sampling}$ seconds a logarithmic curve was fit to the measured data points. This allowed for a sample to be calculated each $t_{sampling}$ second interval based on the

value of the curve at that time instance. Using a logarithmic function was deemed suitable as the temperature rises rapidly initially and slows down over time converging to some thermal equilibrium, according to Newtons Law of cooling and heating [12]. The rate of change in the temperature is extracted as an additional feature, calculated as $\Delta T_i(t) = \frac{T_i(t) - T_i(t-1)}{t_{sampling}}, i \in [m, p]$.

In summary, the pre-processing steps described in this chapter ensure that both the PSD positions and temperature measurements are synchronized, noise-reduced, and enriched with velocity features. Having established this data preparation framework, the subsequent chapter, *Simulation Model*, presents the simulated model inspired by the empiric datasets, used to train and evaluate the performance of the proposed method.

3

Simulation Model

The development of a simulation model for data generation is motivated by the substantial amount of data required to train robust machine learning (ML) models for predictive maintenance (PdM). In addition, the physical machinery is located in a production environment rather than a controlled laboratory setting, which imposes strict operational constraints on allowable runtimes and mechanical loading conditions.

Furthermore, the objective of developing a generalized framework requires access to diverse datasets representing a range of mechanical deviations. Such variations cannot be systematically induced in the physical system without disrupting normal operation. A simulation environment therefore enables controlled and repeatable generation of data under varying conditions, while maintaining consistency with observed system behavior. In this section the developed model is presented.

3.1 Mathematical Model

The model is designed to capture key characteristics observed in the real motor-pump system. Rather than reproducing the system exactly, the objective is to generate representative data that reflects realistic operating behavior and underlying physical trends. This approach enables the development and evaluation of prediction algorithms under controlled conditions, while maintaining consistency with observed system dynamics.

The simulation incorporates key physical effects, including machine dynamics and thermal expansion, which are primary contributors to shaft misalignment over time. In particular, thermal expansion induces differential growth of machine components, leading to shifts in shaft centerlines and resulting in misalignment [13]. Conversely, shaft misalignment introduces additional friction and mechanical losses, increasing heat generation in gears and further exacerbating thermal expansion [14], [10].

Parameters modeling the evolution of the temperature, such as thermal time constants and maximum temperatures, were modeled to mimic the collected temperature data. Machine-specific parameters connecting temperature to PSD measurements are sampled from uniform distributions, while stochastic effects are modeled using Gaussian noise. This parameterization enables the generation of diverse operating conditions, facilitating the development of a generalized prediction model

capable of handling variations across different machines operating under similar environments.

3.1.1 Temperature Dynamics

The temperature model is developed based on Newtons laws of cooling [12], with an additional contribution of thermal energy proportional to the RPM of the motor ($\omega(t)$). Note that $\omega(t)$ only directly effects the motor temperature since this is the primary source of heat generation in the motor, while the pump temperature is influenced mainly through the pumped liquid. The target temperature of motor and pump based on the current $\omega(t)$ of the motor and is modeled as following:

$$T_{m,target}(t) = T_a + (T_{m,max} - (T_{m,max} - T_{m,start})e^{-\zeta_m t} - T_a) \frac{\omega(t)}{2900} \quad (3.1)$$

$$T_{p,target}(t) = \begin{cases} T_a, & \omega(t) = 0 \\ T_a + (T_{p,max} - (T_{p,max} - T_{p,start})e^{-\zeta_p t} - T_a), & \omega(t) \neq 0 \end{cases} \quad (3.2)$$

where ζ_m, ζ_p are the the thermal time constants for the motor and pump respectively. The initial temperatures for the motor and pump $T_{m,start}, T_{p,start}$ are sampled from $\mathcal{U}(25, 30)$ each simulation. Ambient temperature is set to $T_a = 25.2^\circ C$. $T_{m,max}, T_{p,max}$ are the maximum temperature that the motor and pump can reach, as heat generation from the machinery equals the heat dissipation to the environment, these are where specified based on an logarithmic fit to real readings from the physical setup. Where the terminal temperature for the motor and pump is estimated to be $T_{m,max} = 45^\circ C, T_{p,max} = 61^\circ C$.

Temperature update motor and pump:

$$T_m(t) = T_m(t-1) + k_m(T_{m,target}(t) - T_m(t-1)) \quad (3.3)$$

$$T_p(t) = T_p(t-1) + k_p(T_{p,target}(t) - T_p(t-1)) \quad (3.4)$$

$$k_m = \begin{cases} k_{m,heat} & , T_{m,target}(t) - T_m(t-1) > 0 \\ k_{m,cool} & , T_{m,target}(t) - T_m(t-1) \leq 0 \end{cases} \quad (3.5)$$

$$k_p = \begin{cases} k_{p,heat} & , T_{p,target}(t) - T_p(t-1) > 0 \\ k_{p,cool} & , T_{p,target}(t) - T_p(t-1) \leq 0 \end{cases} \quad (3.6)$$

$k_{m,heat} = 0.06, k_{m,cool} = 7 \cdot 10^{-4}$ and $k_{p,heat} = 0.09, k_{p,cool} = 10^{-4}$ accounting for a slower cooling process of the machinery compared to heating the machinery.

The derivatives for each tracked temperature where calculated as following:

$$\Delta T_m(t) = (T_m(t) - T_m(t-1))/\Delta t \quad (3.7)$$

$$\Delta T_p(t) = (T_p(t) - T_p(t-1))/\Delta t \quad (3.8)$$

where the sampling time $\Delta t = 10s$.

3.1.2 Heat Bias and Equilibrium Points

Thermal expansion is a primary driver of structural displacement in rotating machinery. A study on steam turbines demonstrated that temperature differences between ambient and operating conditions lead to material expansion in the casing, directly altering the shaft alignment [13].

In this simulation model the heat expansion is modeled by calculating the difference in temperature based on the current and initial temperatures, and mapping it with a coefficient vector representing the thermal properties of the machinery.

$$v_{exp,m}(t) = \begin{bmatrix} \alpha_m \\ \beta_m \end{bmatrix} (T_m(t) - T_m(0)) \quad (3.9)$$

$$v_{exp,p}(t) = \begin{bmatrix} \alpha_p \\ \beta_p \end{bmatrix} (T_p(t) - T_p(0)) \quad (3.10)$$

α_m, β_m and α_p, β_p are coefficients related to thermal expansion in the system, and are all modeled as parameters proportional to $\alpha_m, \beta_m, \alpha_p, \beta_p \sim \mathcal{U}(10^{-3}, 5 \cdot 10^{-3})$, which are fixed for each simulation.

Based on inspection of empirically collected PSD measurements, the misalignment trajectory follows a curve converging at an equilibrium point. In order to include this behavior into the model, the simulated PSD measurements were driven towards steady-state values, expressed in equations 3.11–3.12.

$$x_{m,eq}(t) = R_m v_{m,exp}(t) + A_{jump} e^{-t/\tau} \quad (3.11)$$

$$x_{p,eq}(t) = R_p v_{p,exp}(t) - A_{jump} e^{-t/\tau} \quad (3.12)$$

$$R_m = \begin{bmatrix} \cos(\theta_m) & -\sin(\theta_m) \\ \sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \quad (3.13)$$

$$R_p = \begin{bmatrix} \cos(\theta_p) & -\sin(\theta_p) \\ \sin(\theta_p) & \cos(\theta_p) \end{bmatrix} \quad (3.14)$$

$A_{jump} = 0.001$ scales the initial movements when the machine is started, and $\tau \sim \mathcal{U}(20, 50)$ is a damping factor for this initial movement. θ_m, θ_p determine the turning of the equilibrium points movement and are sampled from $\mathcal{U}(-\pi/2, \pi/2)$ respectively.

3.1.3 Machine Dynamics

The dynamics of the machine movement are modeled according to the equations described in this section. The model is considering the positions of the motor and pump, $P_m(t)$ and $P_p(t)$, which are pulled towards the equilibrium points, $x_{m,eq}(t)$ and $x_{p,eq}(t)$, of the system (equations 3.11-3.12). The model incorporates the mechanical

effect that the motor has on the pump due to the axial coupling.

$$P_m(t) = 0.95P_m(t-1) + 0.05x_{m,eq}(t) + 0.0002W(t) \quad (3.15)$$

$$P_p(t) = 0.95P_p(t-1) + 0.05x_{p,eq}(t) + C(t-1) + 0.0002W(t) \quad (3.16)$$

$$C(t) = c(P_m(t) - P_p(t)) \quad (3.17)$$

$$W(t) = \gamma W(t-1) + (1-\gamma)\mathcal{N}(0,1) \quad (3.18)$$

where $c \sim \mathcal{U}(0.03, 0.06)$ is a static coupling coefficient representing how the motor and pump influence each other throughout each simulation. The term $W(t)$ introduces auto correlated noise to the system with $\gamma = 0.9$, which is shared between the motor and pump.

3.1.4 PSD Readings

The simulated PSD measurements are modeled to capture difference in tilt, caused by temperature, movement of the machinery and some random drift. The tilt is modeled with the following equations:

$$\theta_{m,tilt}(t) = \theta_{m,tilt}(t-1) + G_m \cdot (T_m(t) - T_m(t-1)) \quad (3.19)$$

$$\theta_{p,tilt}(t) = \theta_{p,tilt}(t-1) + G_p \cdot (T_p(t) - T_p(t-1)) \quad (3.20)$$

with the gains described as $G_m, G_p \sim 10^{-4} \cdot \mathcal{U}(1, 5)$ being constant through each simulation.

The positional readings of the PSD is modeled as following:

$$S_m(t) = P_m(t) - 0.2P_p(t) + L(\theta_{m,tilt}(t) - \theta_{p,tilt}(t)) \quad (3.21)$$

$$S_p(t) = P_p(t) - 0.2P_m(t) + L(\theta_{p,tilt}(t) - \theta_{m,tilt}(t)) \quad (3.22)$$

$$(3.23)$$

where $S_m(t), S_p(t)$ are the measured (x, y) coordinates of the PSD readings and $L = 535mm$ is the distance between the PSD and laser emitters. The magnitude of the PSD readings $d_i(t)$ is calculated as:

$$d_i(t) = \sqrt{S_{i,x}(t)^2 + S_{i,y}(t)^2}, \quad i \in [m, p] \quad (3.24)$$

where $S_{i,x}$ and $S_{i,y}$ are the x and y components of the PSD readings.

Based on the equations presented in equations 3.1-3.24 datasets of sequences were generated, in order to train and evaluate the algorithm.

3.2 Data Generated from Simulation Model

In this section a comparison between the empirical data and simulated data are presented. Additionally, for illustrative purposes, several simulations are provided alongside an evaluation of different standardization methods for the simulated data.

As discussed in 3.1, the aim of the simulated model is to capture the broad characteristics of the physical system and not be a one to one representation. This as the simulated model is generated from a set of random configurations defining the system dynamics, and each simulated sequence aims to represent the same duration as the empiric data. The comparison is presented in 3.1.

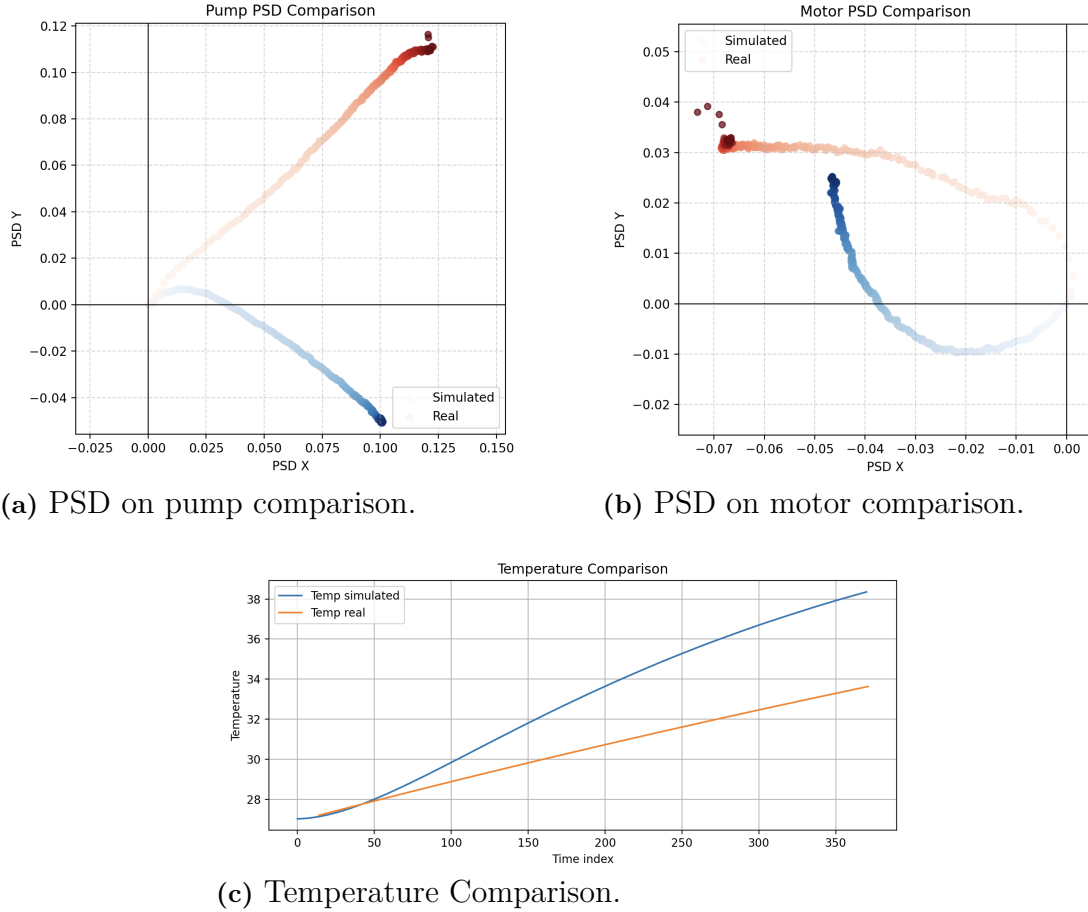


Figure 3.1: Comparison of simulated vs real data.

3.2.1 PSD Simulation

Figure 3.2 shows several simulated trajectories with one change of RPM for both motor PSD and pump PSD. Each simulation is assigned a unique color, and the same color is used in both subplots to indicate corresponding simulations.

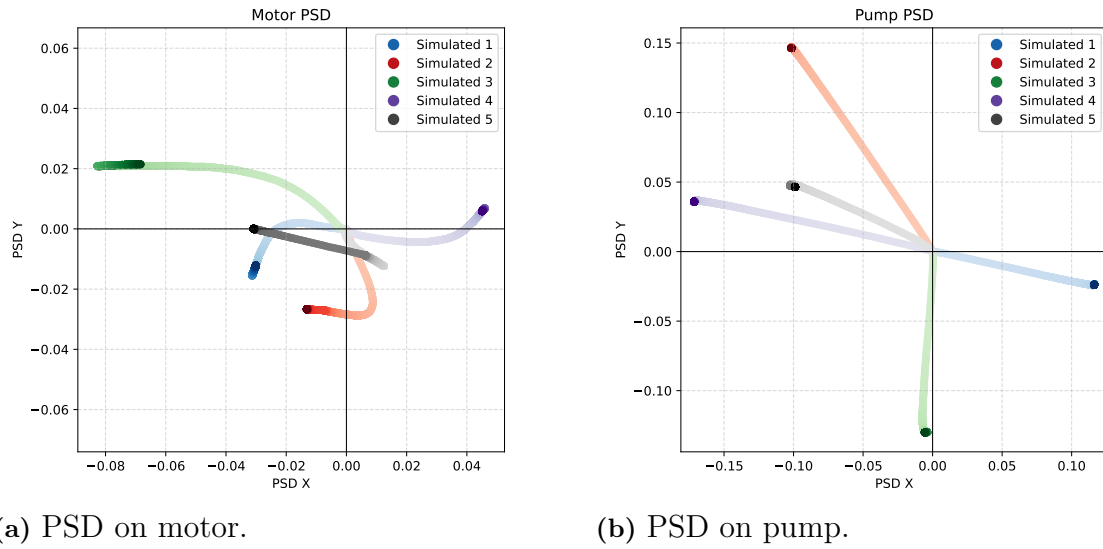


Figure 3.2: Different example PSD trajectories from the simulated machine setup.

3.2.2 Temperature Simulation

The evolution of the temperature in the simulation is majorly impacted by the RPM of the axis, the figures bellow display how the temperature of the motor and pump evolve over time, as well as the RPM of the machine. Simulations depicted in Figure 3.3 are from same simulations as in Figure 3.2.

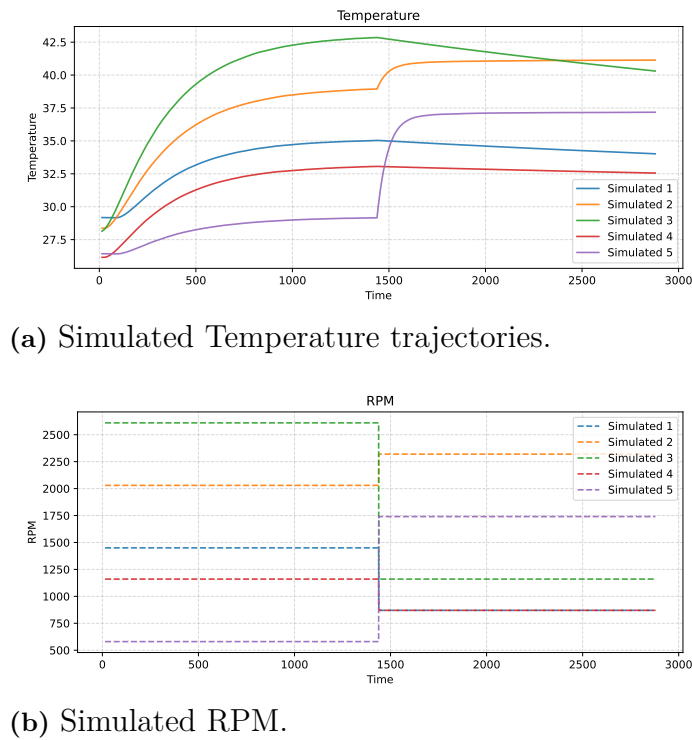


Figure 3.3: Different examples of temperature and RPM characteristics from simulations.

3.2.3 Standardization Methods

Based on the generated data, where some features exhibit low variance, the performance of different normalization methods is presented in Table 3.1.

Table 3.1: Comparison between scaled data metrics, utilizing different scalers fit to 1000 training sequences.

Method	Metric	Features ($T_m, T_p, \Delta T_m, \Delta T_p, \omega, \log(\omega), d_m, d_p$)
MinMaxScaler	Min	[4.39e-08, -4.82e-08, 6.01e-10, 0.0, 8.27e-10, 5.43e-08, -9.15e-14, -4.96e-14]
	Max	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.999]
	Mean	[0.569, 0.966, 0.023, 0.044, 0.516, 0.672, 0.299, 0.571]
	Median	[0.593, 1.0, 0.011, 1.76e-09, 0.555, 0.778, 0.266, 0.597]
RobustScaler	Min	[-1.26, -2.41e04, -1.03, -2.04e-05, -1.0, -1.82, -1.09, -2.39]
	Max	[0.867, 2.04e-05, 87.75, 1.15e04, 0.8, 0.52, 3.00, 1.61]
	Mean	[-0.049, -798.9, 1.08, 517.2, -0.07, -0.24, 0.13, -0.10]
	Median	[3.31e-08, 2.04e-05, 8.82e-12, 0.0, 0.0, -3.49e-08, 2.04e-08, -2.81e-08]
StandardScaler	Min	[-2.09, -7.26, -0.65, -0.27, -1.63, -2.30, -1.72, -3.04]
	Max	[1.58, 0.24, 26.82, 5.93, 1.53, 1.11, 4.03, 2.28]
	Mean	[1.27e-07, -3.31e-07, 9.83e-09, 2.14e-09, -8.54e-08, 1.49e-07, 2.50e-08, -6.78e-08]
	Median	[0.085, 0.248, -0.337, -0.277, 0.124, 0.360, -0.189, 0.135]

In conclusion, the simulation model and standardization methods presented in this chapter provides a controlled environment for generating both nominal and anomalous datasets. With the training and evaluation data thus established, the following chapter, *Algorithm design*, details the specific methodology and architectural choices implemented to detect anomalies within these generated data streams.

4

Algorithm Design

The algorithm consists of two separate modules, a Long-Short term memory (LSTM) for future alignment error prediction, and a Hidden Markov Model with Gaussian mixture emissions (HMM-GMM) for state identification and anomaly detection. The modules are trained and operated individually, in order to evaluate and tune the models inside. The modules can be deployed together as the integrated pipeline, where the HMM-GMM module performs state identification and anomaly detection based on the predictions generated by the LSTM. Figure 4.1 is an illustration of the information pipeline from processed data to anomaly detection.

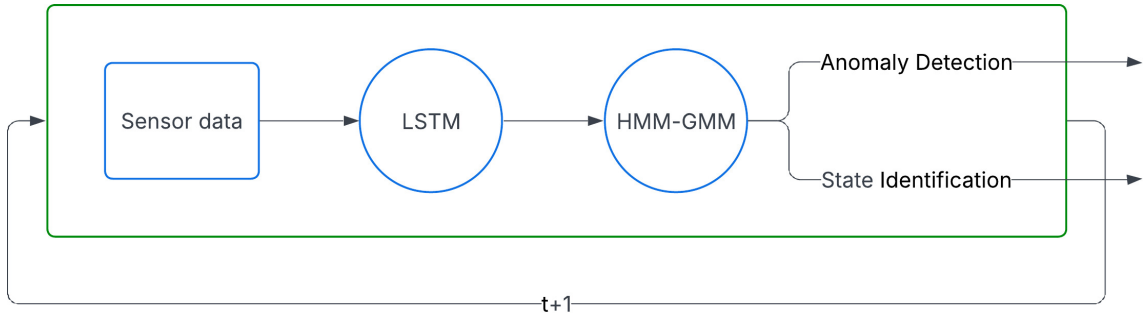


Figure 4.1: Illustration of information pipeline.

4.1 Alignment Prediction

The goal for alignment prediction is to develop an algorithm able to accurately estimate the PSD values magnitude of the offsets, i.e. $d_i = \sqrt{x_i^2 + y_i^2}$, where $i \in [m, p]$. The model architecture explored for this task is a LSTM [15]. The algorithm is trained on data generated from the simulation model described in section 3.1. The simulated data is stored in different sets, training $\mathcal{D}_{\text{train}}$, validation $\mathcal{D}_{\text{val}}$ and test $\mathcal{D}_{\text{test}}$. This in order to evaluate the models performance on unseen data.

The known input sequence length was fixed at $l = 50$ data points to enable the LSTM to capture local temporal dependencies within a defined window. Given a sampling time of $\Delta t = 10s$, this configuration provides the model with a 500-second look-back window. This duration is hypothesized to be sufficient for capturing the relevant system dynamics, under the assumption that older data points possess diminishing predictive value for the current state. The predicted sequence is symmetrically configured to $h = 50$ data points, while misalignment induced by thermal expansion

evolves over time, observed in the empirically collected data illustrated in Figure 3.1, a 500 second horizon provides sufficient lead time for operators or automatic systems to react and deploy necessary countermeasures. Furthermore, employing a fixed-length window ensures constant memory complexity and computational efficiency, avoiding the overhead associated with processing arbitrarily long historical sequences. Figure 4.2 is an illustration of how the LSTM interprets input data and predicted sequence.

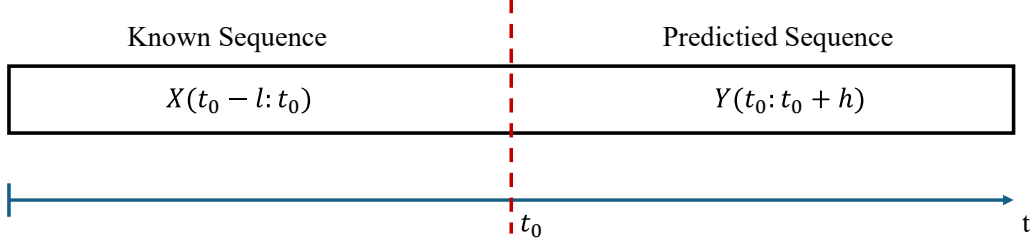


Figure 4.2: Illustration of LSTM input and output information. X denotes the input feature set for the LSTM, while Y represents the predicted output. l represents the length of the known sequence and h represents the length of the prediction horizon.

4.1.1 The LSTM Model

Long Short-Term Memory (LSTM) model architecture is a spin-off from Recurrent Neural Networks (RNN) originally introduced by Hochreiter and Schmidhuber in 1997 [15]. The goal of the LSTM is to resolve vanishing gradient problems that occur when performing backpropagation with RNN using long time series of data [16].

The regular RNN utilized hidden states, $\mathbf{h}_t$, in order to store information over time. The hidden states updates each time step t , producing an output $\mathbf{o}_t$ according to the following equations:

$$\begin{aligned}\mathbf{h}_t &= \phi(U\mathbf{x}_t + V\mathbf{h}_{t-1} + \mathbf{b}_h) && \text{(hidden state),} \\ \mathbf{o}_t &= \psi(W\mathbf{h}_t + \mathbf{b}_o) && \text{(output)}\end{aligned}$$

where $\mathbf{x}_t$ is the input vector, U, V and W are learnable weight matrices for the model, and $\mathbf{b}_h, \mathbf{b}_o$ are biases. ϕ and ψ are the activation functions for the hidden layers and output respectively.

LSTMs complement the RNN by including a cell state, $\mathbf{c}_t$, in addition to the hidden state, $\mathbf{h}_t$. The cell state operates as the long term memory and information in the cell state is updated based on the candidate cell state $\tilde{\mathbf{c}}_t$ controlled by a forget gate, $\mathbf{f}_t$, and an input gate, $\mathbf{i}_t$. An illustration of the LSTM model architecture is displayed

in Figure 4.3.

$$\begin{aligned}
\mathbf{f}_t &= \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) && \text{(forget gate),} \\
\mathbf{i}_t &= \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) && \text{(input gate),} \\
\tilde{\mathbf{c}}_t &= \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) && \text{(candidate cell state),} \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t && \text{(new cell state),} \\
\mathbf{o}_t &= \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) && \text{(output gate),} \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) && \text{(new hidden state).}
\end{aligned}$$

where the W, U terms are learnable weights during training and terms $\mathbf{b}$ are biases [16]. The activation functions used in the LSTM are the $\tanh$ and σ . $\odot$ is the element-wise product operator.

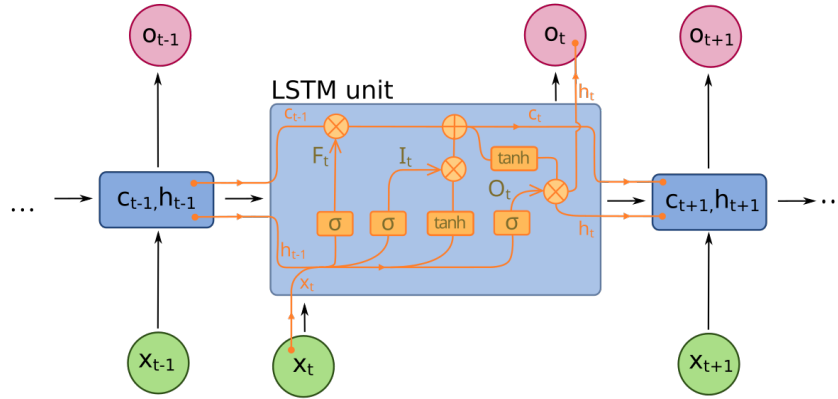


Figure 4.3: Diagram of LSTM architecture [17].

4.1.2 LSTM Prediction Model Implementation

The LSTM model is designed to predict the PSD magnitude of the motor and pump offsets (d_m, d_p) and their respective temperatures (T_m, T_p). The output size scales according to the prediction horizon (h), with specific definitions provided in Table 4.1. Implementation of the LSTM models was done utilizing the `torch.nn.LSTM` module from PyTorch library [18]. During the training phase, simulation data were grouped into batches (Table 4.3) and shuffled to improve generalization and mitigate overfitting [19].

Input features were normalized using MinMaxScaler, a standard practice for recurrent architectures to ensure numerical stability [20]. This choice is specifically motivated by the distributional characteristics observed in Table 3.1. Features such as T_p and ΔT_p exhibit significant skewness, MinMaxScaler effectively maps these non-Gaussian distributions to the range $[0,1]$. By forcing all features into the range $[0,1]$ the inputs are limited to the ranges of the LSTM's internal sigmoid functions, preventing vanishing gradients and ensuring no single feature dominates the weight

update process.

The ADAM optimizer was selected for weight updates. ADAM is a first-order gradient-based optimizer that leverages the benefits of both RMSprop and Momentum to adapt learning rates during training [21].

To ensure robust performance, a model checkpointing strategy was implemented. While the LSTM was trained on the training set $\mathcal{D}_{\text{train}}$, the saving condition was tied to the performance on a disjoint validation set $\mathcal{D}_{\text{val}}$, generated from the same simulation model [22]. The model state corresponding to the lowest validation loss after each epoch was preserved, preventing the network from over-optimizing on the training data's noise.

Table 4.1: LSTM output definition.

Output	Unit	Description
$\hat{d}_m(t+1:t+h)$	mm	Predicted PSD offset magnitude motor
$\hat{d}_p(t+1:t+h)$	mm	Predicted PSD offset magnitude pump
$\hat{T}_m(t+1:t+h)$	$^{\circ}C$	Predicted motor temperature
$\hat{T}_p(t+1:t+h)$	$^{\circ}C$	Predicted pump temperature

4.1.2.1 Tuning Process

After deciding which scaler, loss function and input features provide the best conditions for learning, tuning continued optimizing parameters such as hidden size as well evaluating how the size of the training and validations set affect the performance.

Initial Tuning

The loss functions that are evaluated are Mean Square Error (MSE) and Mean Absolute Error (MAE). MSE and MAE are two of the most common loss functions in machine learning. MSE has its strengths in fast convergence during training, whilst sensitive to errors caused by large outliers, as errors are squared [23]. MAE on the other hand improves on robustness as it minimizes the error across all values equally, with often slower convergence during training. The following equations describes how MSE and MAE are calculated:

$$\text{MSE} : \frac{1}{n} \sum_i^n (y_i - \hat{y}_i)^2 \qquad \text{MAE} : \frac{1}{n} \sum_i^n |y_i - \hat{y}_i|$$

where $\hat{y}_i$ is the i -th prediction with y_i the corresponding true value.

The different combinations of input features that are evaluated in this report are represented in table 4.2.

Table 4.2: LSTM input case definitions for different feature sets, with l representing the known sequence length.

Case 1:		
$T_m(t-l:t)$	$^{\circ}C$	Temperature motor
$T_p(t-l:t)$	$^{\circ}C$	Temperature pump
$\omega(t-l:t)$	RPM	Motor angular velocity
$d_m(t-l:t)$	$[mm]$	PSD offset magnitude motor
$d_p(t-l:t)$	$[mm]$	PSD offset magnitude Pump
Case 2:		
$T_m(t-l:t)$	$^{\circ}C$	Temperature motor
$T_p(t-l:t)$	$^{\circ}C$	Temperature pump
$\log(\omega(t-l:t))$	RPM	Motor angular velocity
$d_m(t-l:t)$	$[mm]$	PSD offset magnitude motor
$d_p(t-l:t)$	$[mm]$	PSD offset magnitude Pump
Case 3:		
$T_m(t-l:t)$	$^{\circ}C$	Temperature motor
$T_p(t-l:t)$	$^{\circ}C$	Temperature pump
$\omega(t-l:t)$	RPM	Motor angular velocity
Case 4:		
$T_m(t-l:t)$	$^{\circ}C$	Temperature motor
$T_p(t-l:t)$	$^{\circ}C$	Temperature pump
$\log(\omega(t-l:t))$	RPM	Motor angular velocity
Case 5:		
$T_m(t-l:t)$	$^{\circ}C$	Temperature motor
$T_p(t-l:t)$	$^{\circ}C$	Temperature pump
$\Delta T_m(t-l:t)$	$^{\circ}C/s$	Temperature derivative motor
$\Delta T_p(t-l:t)$	$^{\circ}C/s$	Temperature derivative pump
$\log(\omega(t-l:t))$	RPM	Motor angular velocity
$d_m(t-l:t)$	$[mm]$	PSD offset magnitude motor
$d_p(t-l:t)$	$[mm]$	PSD offset magnitude Pump
Case 6:		
$T_m(t-l:t)$	$^{\circ}C$	Temperature motor
$T_p(t-l:t)$	$^{\circ}C$	Temperature pump
$\Delta T_m(t-l:t)$	$^{\circ}C/s$	Temperature derivative motor
$\Delta T_p(t-l:t)$	$^{\circ}C/s$	Temperature derivative pump
$d_m(t-l:t)$	$[mm]$	PSD offset magnitude motor
$d_p(t-l:t)$	$[mm]$	PSD offset magnitude Pump

Parameter Tuning

To improve the performance of the specified case, loss function and scaling method, the model is further tuned with different model parameters. Table 4.3 displays the LSTM parameters and hidden sizes that are evaluated. The effect of different training and validation set size will be evaluated, this in order to give an approximation of the quantity of real collected data required to retain a model with similar prediction

performance, see Table 4.4 for set size definitions.

Table 4.3: LSTM dimensions.

Parameter	Dimensions	Description
Batch Size	128	Number of simultaneous computations
Hidden Size	[32, 64, 128, 256]	Number of hidden states
Layers	2	Number of stacked LSTM modules
Input Size	$N_f \cdot l$	Number of features (N_f) times length of known sequence
Output Size	$4 \cdot h$	Prediction dimensions times horizon length

Table 4.4: Set size definitions.

Set	Sizes	Description
$\mathcal{D}_{\text{train}}^N$	$N = [100, 500, 1000]$	Number of generated sequences used for training
$\mathcal{D}_{\text{val}}^N$	$N = [100, 500, 1000]$	Number of generated sequences used for validation
$\mathcal{D}_{\text{test}}^N$	$N = 100$	Number of generated sequences used for test

4.2 State Identification and Anomaly Detection

To identify operating states across different types of machinery, the proposed state identification algorithm must be highly adaptable, as different motors, pumps, fans, and system configurations exhibit varying dynamics and operate under diverse environmental conditions. Because of these varying dynamics and operating environments, an unsupervised learning approach is suitable, as information about state labels is limited. Several different methods can be used for this purpose, including density-based, centroid-based, hierarchical, and distribution-based clustering techniques [24]. Among these, distribution-based approaches are particularly relevant, as they model the data as a mixture of underlying probability distributions, allowing for probabilistic interpretation of cluster membership.

The method used is a Hidden Markov Model with Gaussian mixture emissions (HMM-GMM), which has previously been applied in machinery fault diagnosis and condition monitoring [25], [26].

To address the need for adaptability, an initial operating period is assumed to represent healthy system behavior for each specific setup. Data collected during this period is used to learn representative operating states that characterize normal conditions. These learned states can then be used as a reference for future observations, enabling the detection of deviations that may indicate anomalies. This is a common approach in unsupervised learning [27]. To avoid computational issues the log-likelihood is used instead of likelihood.

4.2.1 Hidden Markov Model for State Transitions

The Hidden Markov Model (HMM) is a statistical framework based on Markov chains consisting of a finite number of states [25]. An HMM is characterized by two sets of probabilities for each hidden state: the transition probabilities between states and the observation (emission) probabilities. Following similar notation in [25], [26], the transitions can be described as:

$$\mathbf{T} = \begin{bmatrix} t_{11} & \cdots & t_{1N} \\ \vdots & \ddots & \vdots \\ t_{N1} & \cdots & t_{NN} \end{bmatrix}, \quad (4.1)$$

where N denotes the number of states, and t_{ij} represents the transition probability from state i to state j . The observations can be expressed as:

$$\mathbf{B} = \{b_j(\mathbf{O}_t)\}_{j=1}^N \quad (4.2)$$

where b_j denotes the emission probability of observation O_t given state j . In the case of Gaussian Mixture Model (GMM), each emission is modeled as a weighted sum of Gaussian components (see Section 4.2.2). Additionally, the initial starting state probability is defined as:

$$\pi_i = P(q_1 = S_i) \quad (4.3)$$

where q_t is the hidden state at time t and S_i is the state where $1 \leq i \leq N$. The model can thus be expressed as:

$$\lambda = \{T, B, \pi\}. \quad (4.4)$$

In practice, the number of hidden states and Gaussian mixtures significantly affects model performance [25]. A lower number of states provides a coarser approximation of the data, while a higher number increases the risk of overfitting. Additionally, increasing the number of states or mixtures may lead to numerical issues such as singular covariance matrices, especially when the data exhibits low variance or strong linear correlations. Since the inverse of a singular covariance matrix does not exist [22], this makes the E-step in the GMM EM algorithm infeasible. To mitigate these issues, several feature sets are evaluated, and the number of model components is selected using the Bayesian Information Criterion (BIC) [22], [25]:

$$BIC = -2 \cdot \log \text{likelihood} + k \cdot \log(n) \quad (4.5)$$

where k is the number of independent parameters and n is the number of observations.

Baum–Welch Algorithm (Forward–Backward)

The model is trained using the Baum–Welch algorithm, a specific application of the Expectation–Maximization (EM) algorithm that maximizes the log-likelihood $P(O|\lambda)$. The following variables and equations are based on the derivations in [28].

First, the forward variable $\alpha_t(i)$ represents the probability of the partial observation sequence up to time t and state is $q_t = S_i$:

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = S_i | \lambda) \quad (4.6)$$

Similarly, the backward variable $\beta_t(i)$ is defined as the probability of the remaining observation sequence given state S_i at time t :

$$\beta_t(i) = P(O_{t+1}, \dots, O_T | q_t = S_i, \lambda) \quad (4.7)$$

Both variables are calculated recursively. Combining them allows for the calculation of $\gamma_t(i)$, the probability of being in state S_i at time t given the entire observation sequence:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) \quad (4.8)$$

To update the state transitions, the variable $\xi_t(i, j)$ is used. This represents the probability of being in state S_i at time t and transitioning to state S_j at time $t + 1$:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad (4.9)$$

Viterbi Algorithm

Once the HMM is trained, the Viterbi algorithm is employed to determine the most probable sequence of hidden states [25]:

$$q_{1:T}^* = \arg \max_{q_{1:T}} P(q_{1:T}, O_{1:T} | \lambda) \quad (4.10)$$

4.2.2 Gaussian Mixture Emissions

The equations in this section are derived from [28]. A Gaussian Mixture Model (GMM) is used to represent complex data distributions by combining multiple Gaussian components. In the context of an HMM, see Section 4.2.1, the likelihood of an observation O_t being emitted while in state j is defined as:

$$b_j(O_t) = \sum_{k=1}^K w_{jk} \mathcal{N}(O_t | \mu_{jk}, \Sigma_{jk}) \quad (4.11)$$

where w_{jk} , μ_{jk} , and Σ_{jk} represent the weight, mean, and covariance of the k -th mixture component in state j . The model parameters are iteratively refined using the Expectation-Maximization (EM) algorithm [22], [28].

E-step

In the E-step, the model calculates the responsibility of each Gaussian component. This represents the probability that a specific component k was responsible for generating the observation O_t at time t , given that the system is in state j :

$$\gamma_{tk}(j) = \gamma_t(j) \cdot \frac{w_{jk} \mathcal{N}(O_t | \mu_{jk}, \Sigma_{jk})}{\sum_{l=1}^K w_{jl} \mathcal{N}(O_t | \mu_{jl}, \Sigma_{jl})} \quad (4.12)$$

M-step

In the M-step, the model parameters are updated to better fit the data based on the probabilities calculated in the E-step. The new means, covariances, and weights are re-estimated by taking weighted averages of all observations. This ensures that each component's parameters move closer to the data points it most likely generated, thereby maximizing the overall likelihood of the model [28].

4.2.3 Model Implementation

The HMM-GMM should be able to identify states and abnormal behavior base on laser measurements and temperature. To accomplish this, a model is created that is capable of taking external temperature sensor information and laser measurements as inputs, denoted as the full model. Furthermore, in the absents of laser measurements during operation, the model should still be able to preform on just temperature reading. A second model is therefore created from the full model, where the laser measurement data is marginalized over, denoted as the marginalized model. By keeping the laser measurements during training for the marginalized model, it provides the benefit of further separating states related to misalignment.

The marginalization is possible because of the usage of Gaussian emissions, which preserves Gaussianity [22]. The marginalization is calculated as:

$$P(X = x) = \sum_y P(X = x, Y = y) \quad (4.13)$$

where X represents data from external temperature sensors and Y represents data from the laser measurement units. Assuming that similar input vectors correspond to consistent system behavior, the marginalized model can identify operating states and abnormal behavior based solely on external temperature data. Meaning that as long as deviations in temperature corresponds to moment in the alignment, the different operating conditions can be identified.

The model is implemented using the *hmmlearn* library [29]. The number of training iterations was set to 200, as this was found to provide an appropriate balance between model performance and computational efficiency. Additionally, a fixed seed was used for state initialization to ensure reproducibility. Prior to model training, all features are standardized using `StandardScaler`. As shown in Table 3.1, this method effectively centers the data with a mean near zero and maintains a consistent scale across all features. Unlike `MinMaxScaler`, which compresses data into a narrow range when outliers are present, or `RobustScaler`, which resulted in extreme magnitudes for temperature differences, `StandardScaler` provides a balanced distribution.

4.2.4 Model Evaluation

To verify the accuracy of the algorithm, tests where conducted on both empirical and simulated data. Different feature sets where evaluated to assess the preference

and stability, and the ability reproduce stats from the full model. The empirical data was paired according to 2.4 with a resampling time of 5 seconds and the simulated data sequence was set to capture several operating conditions.

Feature set Creation

To find the optimal features, feature configuration and the number of features to train the model on, tests were done to evaluate the state identification accuracy and anomaly detection ability. The feature-sets was created through different combinations and variations of PSD values (raw component values and combined magnitudes), temperature (raw temperature from motor and pump and merged combinations) and expansion velocities and temperature derivatives. Considerations such as the number of dimensions and data separability was accounted for when designing the feature-sets. A baseline feature set was chosen as it was hypothesized to provide good performance and capture the systems behavior. The feature set can be seen in table 4.5.

Table 4.5: Feature set example.

Feature Set	Description
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}, \sqrt{\Delta T_m^2 + \Delta T_p^2}\}$	Magnitude of misalignment motor PSD, Magnitude of misalignment pump PSD, Combined temperature magnitude, Combined temperature difference magnitude

State Construction and Identification

The Bayesian Information Criterion (BIC) [22] is used to select the number of states and Gaussian mixtures, as it can avoid overfitting [25]. The available dataset based on empirical measurements exhibits relatively simple behavior, with operation at near-constant rotational speed and a high degree of correlation between measured variables. Similarly, the simulations were constructed to cover few operational scenarios. Therefore, the model complexity is kept low. For each feature set, models with 2–7 hidden states and 1–3 Gaussian mixtures per state are evaluated using BIC. Each configuration is tracked, and any model that results in numerical instability is classified as a failure. Such failures may arise due to highly correlated or redundant features, which can lead to ill-conditioned covariance matrices, but can also be caused by insufficient data assigned to individual mixture components. A minimum covariance value is set to increase robustness.

Beyond numerical stability, the models are evaluated on their ability to distinguish between distinct operational states, such as heating, cooling, and stable operation. To ensure practical utility, the stability of these states is assessed, as excessive state switching indicates a lack of separation between Gaussian components.

To quantify this performance, a metric called Match-Ratio is introduced. This metric measures the consistency between the marginalized HMM-GMM and its respective full model. Furthermore, the models' real-time identification capabilities are evaluated by comparing pre-identified states from the complete dataset with those generated in a simulated real-time scenario. This dual approach allows for an assessment of how well the models maintain state-reconstruction accuracy when sensor data is restricted or arriving sequentially.

Anomaly Detection

To identify abnormal behavior, the HMM-GMM model calculates the log-likelihood of the sequence of observations of a set size. Compared to solely relying on GMM, this gives the likelihood of the entire sequence being observed compared to only point wise observations. This make the system robust towards temporary anomalies and outliers. Since the algorithm is trained solely on healthy system behavior, the anomaly threshold is set to classify as anomalous any points that lie beyond three standard deviations of the observed data. To test the model, abnormal behavior are introduced to the data in the form of abnormal temperature and PSD trajectories.

4.3 Pipeline

The complete architecture consists of an HMM-GMM for anomaly detection and an LSTM for PSD magnitude and temperature estimation. Although these models function independently, the integrated pipeline leverages both. The LSTM provides an estimate of the features, which is then used as input to the regular HMM-GMM model for anomaly detection. The method provides the HMM-GMM with additional information, but introduces a stronger dependency on the performance of the LSTM.

Although alternative approaches, such as autoencoder-based methods (e.g., [30]) and other clustering-based techniques [24], may offer strong performance, they often lack the same level of interpretability. For the purposes of this thesis, where understanding and diagnosing system behavior is of central importance, the proposed pipeline provides a more suitable balance between performance and explainability.

4.3.1 Testing of Pipeline

The method in 4.3 is evaluated based on its ability to predict anomalies ahead of time, thereby assessing the effectiveness of the combined LSTM and HMM-GMM framework for predictive anomaly detection. Abnormalities are introduced in the input data for the LSTM which then are propagated through to the HMM-GMM through its prediction for evaluation of the pipeline.

In summary, this chapter has outlined the architectural design of the HMM-GMM anomaly detection and operating states identification as well as the LSTM for predictions framework. Having defined the algorithm's structure and the evaluation

metrics, the subsequent chapter, *Results*, presents the findings and performance evaluations obtained from running the algorithm on the simulated data.

5

Results

This chapter presents the key findings from the evaluation of the individual modules and the pipeline as a whole. First, the performance of the LSTM network is analyzed with focus on hyperparameter tuning and prediction accuracy. Subsequently, the HMM-GMM framework is evaluated based on its state identification capabilities, feature set utilization, and anomaly detection performance. Finally, the results conclude with an assessment of the integrated pipeline, demonstrating how these combined modules function in a unified system.

5.1 LSTM Performance and Prediction Accuracy

This section evaluates the performance of the LSTM module across various training parameter configurations and input feature sets. The specific input feature configurations are detailed in Table 4.2 and are referred to as *Case X* throughout this analysis. Furthermore, the impacts of the loss function selection, the hidden size of the model, and dataset volume are examined.

To quantify model performance, the Mean Absolute Error (MAE) and Mean Squared Error (MSE) are utilized. These metrics are calculated based on the prediction error, the residual difference between the predicted and actual values, across all sequences in the test set. To ensure a robust assessment of generalization, the test set remains entirely disjoint from the training and validation sets. This separation provides an unbiased metric of how the model performs when processing previously unseen data.

5.1.1 LSTM Tuning

In order to determine which input case and training loss function performed best at predicting future values, numerous models were trained with the same structural dimensions, seen in Table 5.1, only changing the input and loss function. This is in order to isolate the impact the input parameters have on the predictor. Subsequently, the volume of the training and validation datasets are evaluated, as well as the hidden size of the LSTM model. Working towards a final tuning of the model.

Table 5.1: Baseline LSTM model configuration.

Parameter	Value
Batch Size	128
Hidden Size	256
Prediction horizon (h)	50
Layers	2
Known Sequence (l)	50
Set sizes: $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$	100,100,100
Max Epochs	30

Case and Loss Function Evaluation

In order to evaluate which training loss function and Case of input features displays the best performance, multiple LSTM was trained utilizing different combinations of these. In Figure 5.1-5.3 illustrations of the trajectories predicted by the trained models compared to the actual trajectory of the system for a set of test sequences, with the prediction 50. Note that in the figure legends models trained with MAE are labeled L1Loss.

From the displayed trajectories it is clear that models for Case 3 and 4 struggles to predict PSD offsets for both motor and pump, indicated by the large offsets in the all figures, implying that the prediction tasks require more information. One more interesting thing to note with Case 3 and 4 is that the predicted offsets for the motor and pump are similar between test sequences, approximately 0.07 and 0.13 respectively before the RPM change.

By visual inspection for the prediction sequences it is observed that models trained with MAE outperforms models trained with MSE, this is also reflected in Table 5.2 where models trained with MAE excels over models trained with MSE. Marked in bold are the best performing models for the MSE and MAE criterion respectively.

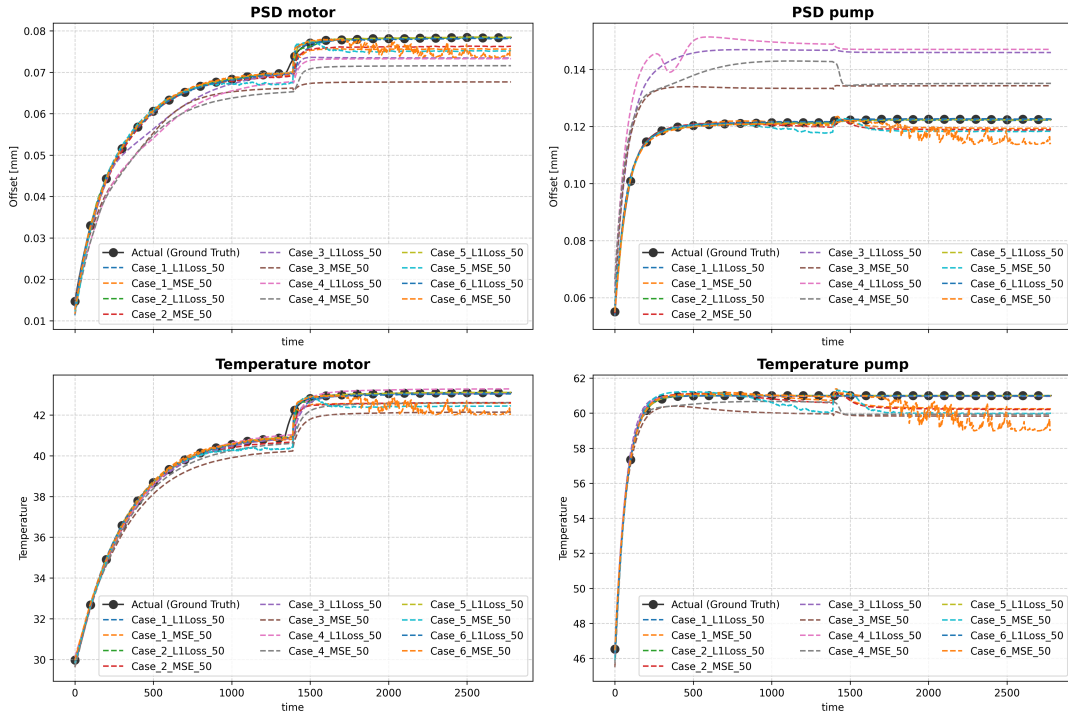


Figure 5.1: LSTM model prediction comparison based on input case, for sequence 1 from $\mathcal{D}_{\text{test}}$.

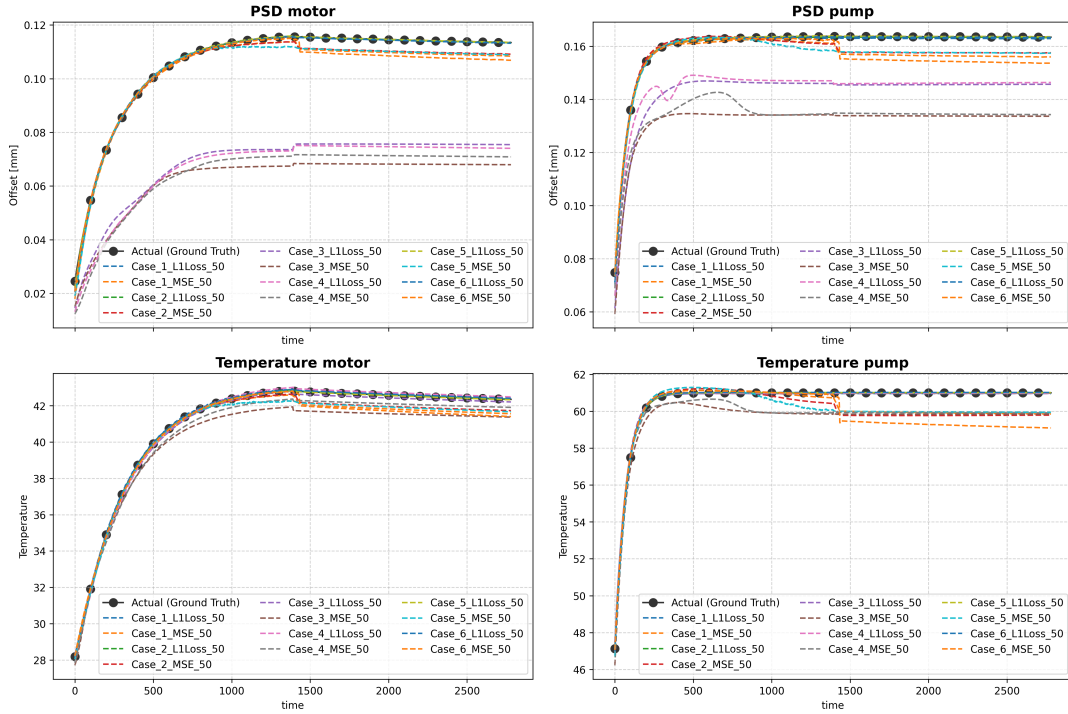


Figure 5.2: LSTM model prediction comparison based on input case, for sequence 2 from $\mathcal{D}_{\text{test}}$.

5. Results

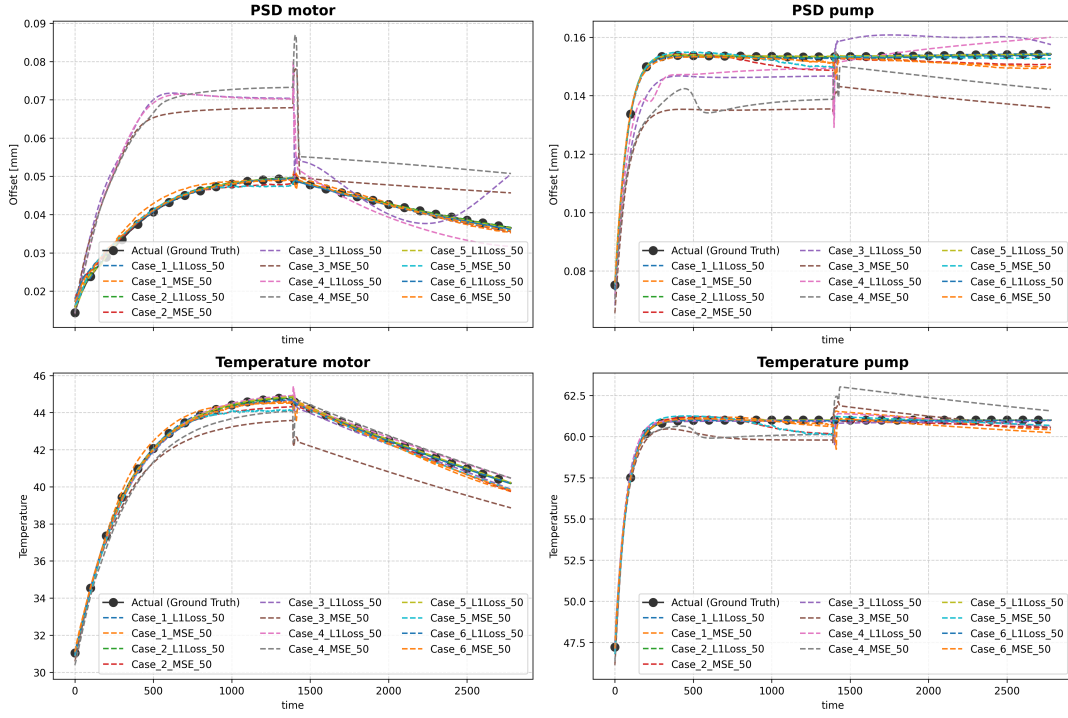


Figure 5.3: LSTM model prediction comparison based on input case, for sequence 3 from $\mathcal{D}_{\text{test}}$.

Table 5.2: Experimental Results of MAE and MSE by Case and Loss Function.

Case	Loss Function	Horizon	MAE	MSE
Case 1	MAE	50	0.024300	0.025905
Case 1	MSE	50	0.163675	0.129962
Case 2	MAE	50	0.021217	0.026997
Case 2	MSE	50	0.183369	0.148602
Case 3	MAE	50	0.080327	0.052032
Case 3	MSE	50	0.401298	0.404488
Case 4	MAE	50	0.073138	0.052478
Case 4	MSE	50	0.265180	0.211442
Case 5	MAE	50	0.020814	0.026465
Case 5	MSE	50	0.213899	0.174669
Case 6	MAE	50	0.030349	0.023151
Case 6	MSE	50	0.202110	0.234944

The set-up with MAE and cases 5 and 6 are further developed in upcoming sections. Evaluating how varying the amount of training data affects the models performance.

Dataset Size

In order to evaluate how well the developed model performs when trained different sizes of datasets $\mathcal{D}_{\text{train}}^N$, see Table 4.4 for definitions. In table 5.3 a comparison between the models performances on a set of 100 test sequences are displayed, using MAE and MSE as performance metric. Inspecting the data further it is possible to see trends between more data and better performance. This as both MAE and MSE decrease for the test set with an increased number of sequences in $\mathcal{D}_{\text{train}}^N$ and $\mathcal{D}_{\text{val}}^N$ sets.

Table 5.3: Performance for input features Case 5 and 6, minmax scalers and MAE loss function, for different sizes of $\mathcal{D}_{\text{train}}^N$ and $\mathcal{D}_{\text{val}}^N$, with $\mathcal{D}_{\text{test}}$ of size 100 sequences.

Case	Dataset Size $\mathcal{D}_{\text{train}}^N, \mathcal{D}_{\text{val}}^N$	MAE	MSE
Case 5	100/100	0.020814	0.026465
Case 6	100/100	0.030349	0.023151
Case 5	500/500	0.020359	0.026414
Case 6	500/500	0.013399	0.025703
Case 5	1000/1000	0.008232	0.019496
Case 6	1000/1000	0.011680	0.019581

The the model utilizing Case 5 with $\mathcal{D}_{\text{train}}^N, \mathcal{D}_{\text{val}}^N$ of size 1000/1000, marked in bold, retains a slight edge in MAE and MSE over the model trained with Case 6. Thus tuning continues with Case 5.

LSTM Hidden Size Tuning

In order to futher tune the models for best performance, experiments with different Hidden Sizes where conducted, using Case 5 trained with MAE loss function. Table 5.4 is the results in regard to MSE and MAE performance of the trained models. It is possible to draw conclusions that with a larger number of hidden parameters the model manages to predict more accurately.

Table 5.4: Experimental Results: Performance dependent on Hidden Size of LSTM model.

Case	Loss Function	Hidden Size	MAE	MSE
Case 5	MAE	32	0.019252	0.021131
Case 5	MAE	64	0.018199	0.020241
Case 5	MAE	128	0.008502	0.019474
Case 5	MAE	256	0.011855	0.019708
Case 5	MAE	512	0.007709	0.019510

The model with Hidden Size 512 shows the strongest performance, and is considered the final tuned model. Which is evaluated in the upcoming section.

5.1.2 Performance of tuned model

In Table 5.5 the final configuration for the LSTM model, and in Figure 5.4 - 5.6 are the predicted outputs for the model performed on sequences from $\mathcal{D}_{\text{test}}$. Notable for the predicted sequences is that the model performs very differently based on the sequence, with notable bias towards an higher initial temperature in the motor visible for the sequences displayed in Figure 5.5 and 5.6. Moreover, noteworthy is the predictors inaccuracy in prediction of the offsets round the instances of the RPM change occurring at time instance 1440.

Table 5.5: LSTM model configuration after tuning.

Parameter	Value
Input features	Case 5
Batch Size	128
Hidden Size	512
Prediction Horizon (h)	50
Layers	2
Known Sequence (l)	50
Set sizes: $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$	1000,1000,100
Max Epochs	30

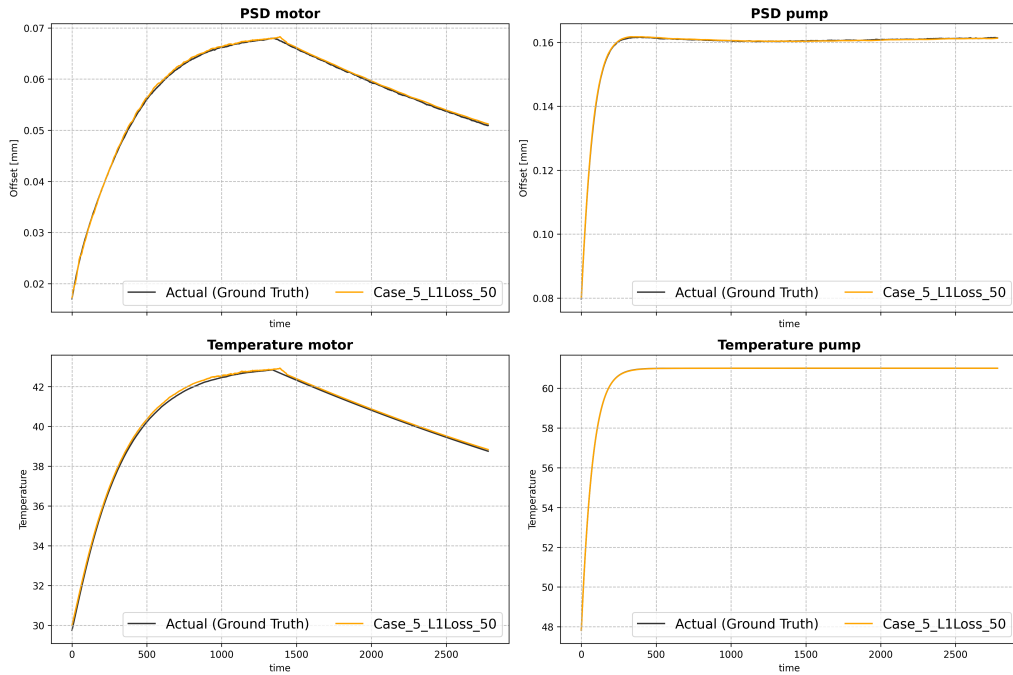


Figure 5.4: Final model estimation on sequence from $\mathcal{D}_{\text{test}}$.

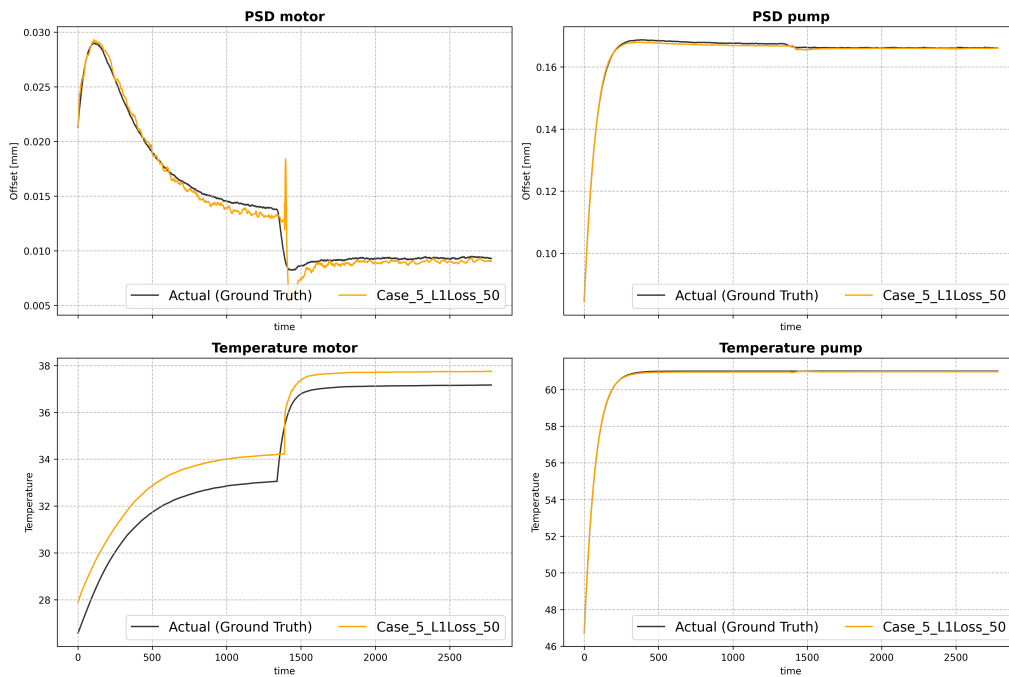


Figure 5.5: Final model estimation on sequence form $\mathcal{D}_{\text{test}}$.

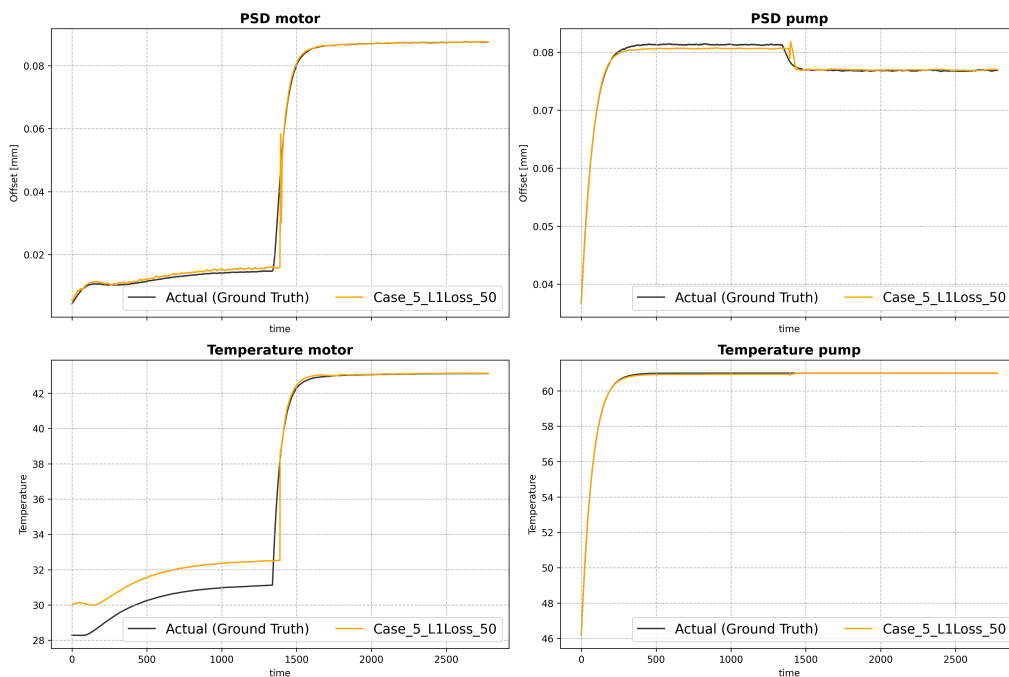


Figure 5.6: Final model estimation on sequence form $\mathcal{D}_{\text{test}}$.

Utilizing this tuned LSTM module, the subsequent chapter evaluates the HMM-GMM modules efficacy in state identification and anomaly detection. The complete pipeline is ultimately assessed in Section 5.3, verifying its capacity for concurrent prediction and anomaly detection.

5.2 HMM-GMM Evaluation and State Identification

Key results from the performance evaluation are summarized in the tables and text below. As a baseline the feature set $\{d_m, d_p, \sqrt{T_m^2 + T_p^2}, \sqrt{\Delta T_m^2 + \Delta T_p^2}\}$ was selected for the primary evaluation. The tests are conducted on the simulated data presented in Figures 5.7, an arbitrary simulation of a system operating under different condition. The simulation consists of several heating, cooling and stable operation sections, especially seen in the development of the motor temperature. It is a desirable outcome for the algorithm to correctly identify and separate these different types of behaviors into states.

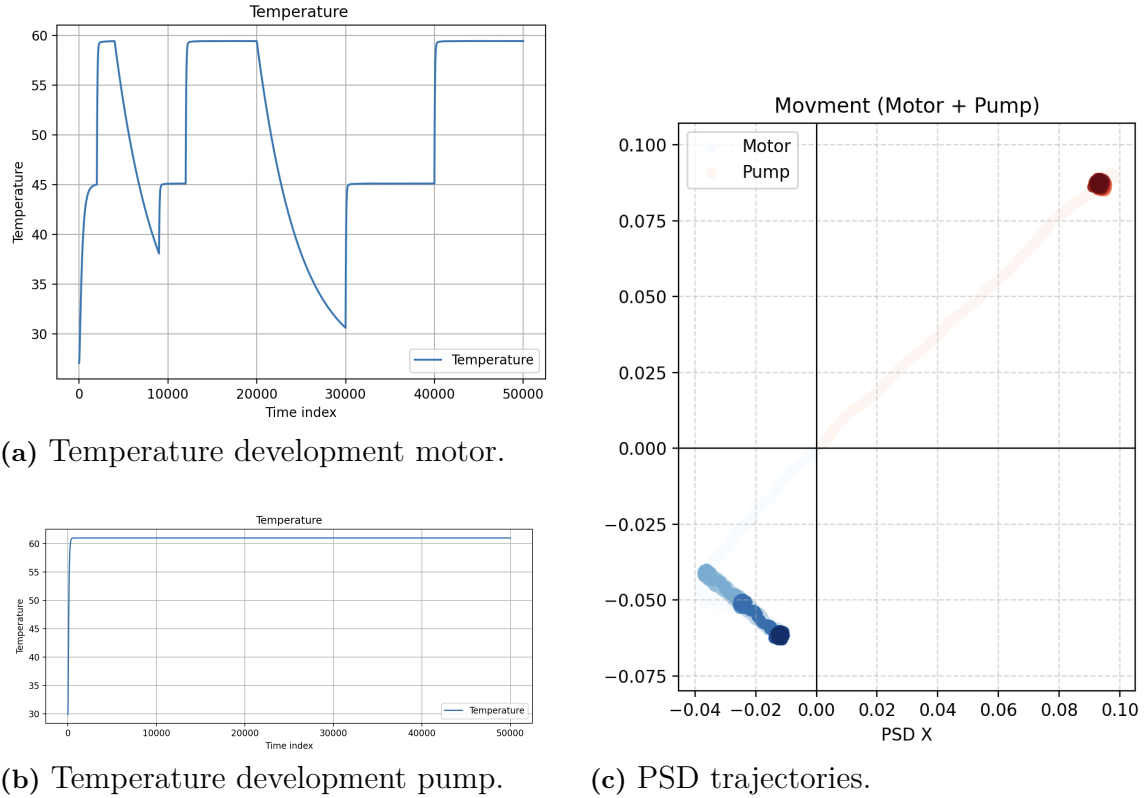


Figure 5.7: Simulated results showing temperature and alignment behavior.

5.2.1 Baseline Model Stability

Table 5.6 illustrates the fundamental trade-off between numerical stability and model resolution. By varying the minimum constraint on the eigenvalues of the covariance matrix and the number of training samples, the analysis highlights how the Bayesian Information Criterion (BIC) adapts the state complexity based on the imposed numerical requirements. This selection demonstrates the conditions under which the model remains robust while still capturing the system dynamics

Table 5.6: Summary of model performance for selected test configurations.

Samples	Min. Eig. Cov	States/Mixtures	Numerical Failures	Log-likelihood
50 000	10^{-8}	7/2	1/18	388 719
50 000	10^{-7}	3/2	9/18	311 085
50 000	10^{-6}	2/1	17/18	924
50 000	10^{-5}	2/1	17/18	-36 540
30 000	10^{-8}	5/3	1/18	262 158
30 000	10^{-7}	5/3	3/18	262 158
30 000	10^{-6}	7/1	13/18	110 163
30 000	10^{-5}	3/1	16/18	31 984
10 000	10^{-8}	6/3	1/18	80 957
10 000	10^{-7}	7/2	4/18	79 113
10 000	10^{-6}	3/2	13/18	57 997
10 000	10^{-5}	3/2	13/18	57 997

The results in Table 5.6 indicate a general correlation between the covariance constraint and the resulting model complexity. As the minimum covariance is increased from 10^{-8} to 10^{-5} , the number of states selected by BIC tends to decrease, while the number of numerical failures increases. However, this trend is not strictly consistent across all training lengths. A value of 10^{-7} is chosen as it provides a favorable balance between capturing system dynamics (as seen in the log-likelihood) and the number of states and mixtures created.

5.2.2 State Construction and Reconstruction Ability

This section evaluates the state reconstruction capabilities of the marginalized versus the full model using the baseline feature set. Furthermore, the models state creation and transition behaviors are presented.

Internal State Consistency

The internal consistency of the marginalized models was evaluated by re-applying the training data. As shown in Table 5.7, the marginalized model exhibits high diagonal dominance. Furthermore, across various training lengths, the observed match ratio between the full and marginalized models ranges from 85.79% to 99.79%.

Table 5.7: Confusion Matrix for internal state consistency (50 000 samples). Row represent the full model and the columns represent the marginalized model.

		Marginalized model		
		State 1	State 2	State 3
Full model	State 1	30 310	3	0
	State 2	91	6 511	4
	State 3	33	476	12 571

This shows that the marginalized model is capable of identifying the same states as the full model.

State Behavior of the HMM-GMM Model

To evaluate the models ability to identify distinct operational states and establish logical transitions, the frequency of state changes was investigated. To illustrate how the states are created and segmented, the motor temperature is presented with color-coded regions corresponding to each identified state (Figure 5.8). This variable is chosen as a representative proxy, as it is used to drive the changes in other simulated parameters and thus effectively reflects the behavior of the entire system.

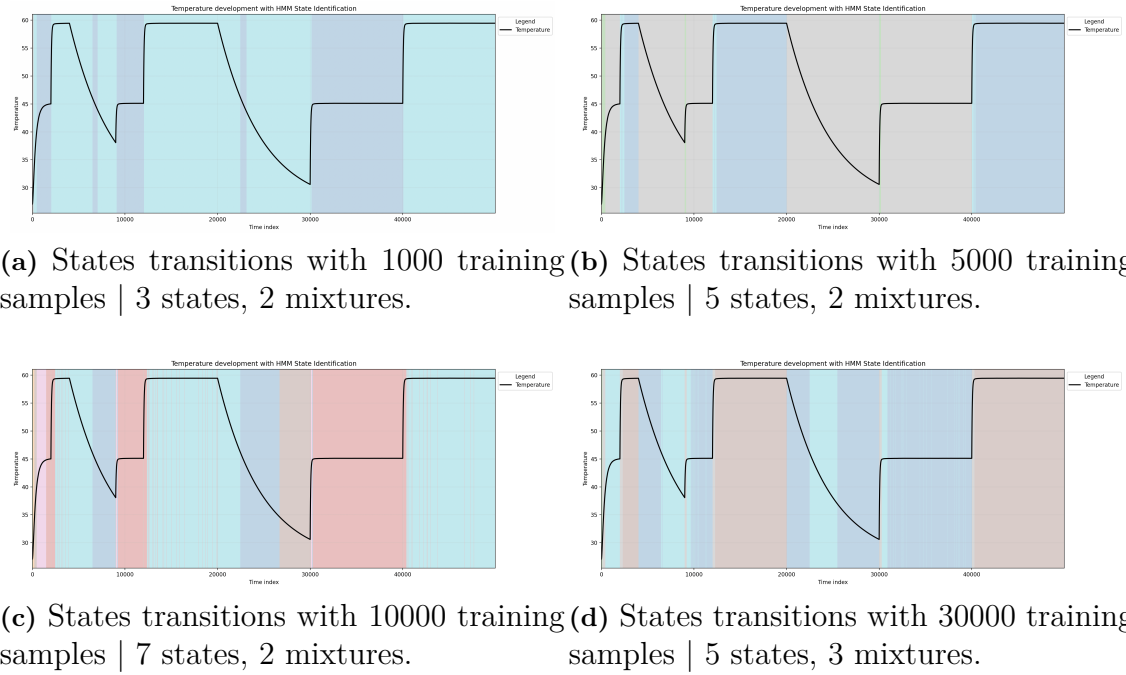


Figure 5.8: State transitions when model training is conducted on different data lengths.

At longer training intervals, the model is prone to high frequent state switching. It can also be observed that the models find it difficult to accurately identify operational states such as heating and cooling.

Real-time state identification for the baseline feature set showed high precision; the full model reached a match ratio of 99.73%, outperforming the marginalized model, which identified 98.87% of the states correctly.

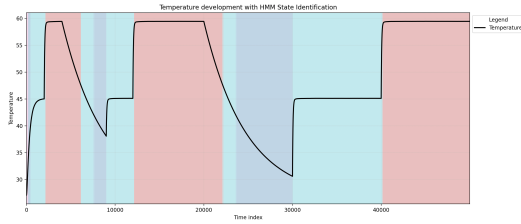
5.2.3 Feature Set Comparison

The feature sets presented in Table 5.8 represent a subset of the total configurations evaluated during the study. Specifically, these sets were selected as they consistently provided stable models across varying training sample sizes. The primary objective is to evaluate and compare the ability of different feature combinations to maintain distinct state separation. The Match Ratio indicates how much the marginalized and full model agrees on the operation states.

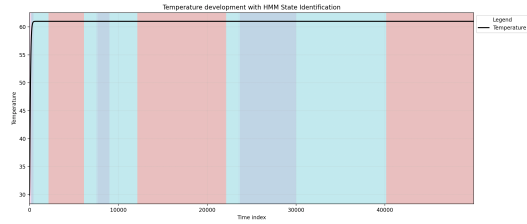
Table 5.8: Consistency and performance metrics of the models across different feature sets (50 000 samples).

Feature Set	States/ Mixtures	Failures	Match Ratio
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}, \sqrt{\Delta T_m^2 + \Delta T_p^2}\}$	3/2	9	98.79%
$\{S_m^2, S_p^2, \sqrt{T_m^2 + T_p^2}, \sqrt{\Delta T_m^2 + \Delta T_p^2}\}$	6/2	7	98.76%
$\{S_m^2, S_p^2, \sqrt{T_m^2 + T_p^2}\}$	6/2	6	93.66%
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$	6/1	10	80.29%

As shown in Table 5.8, feature sets that incorporate temperature dynamics demonstrate a higher capacity for state separation, with match ratios exceeding 98% for both models. The choice of feature set also influences the state transition behavior. Specifically, the set $S_m^2, S_p^2, \sqrt{T_m^2 + T_p^2}$ with 30,000 training samples resulted in consistent state placement and reduced high-frequency state switching, as illustrated in Figure 5.9.



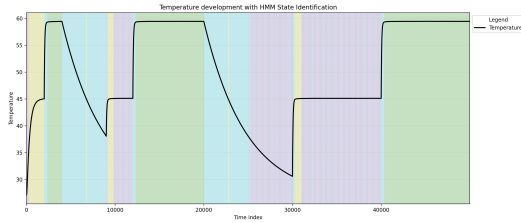
(a) Motor temperature development.



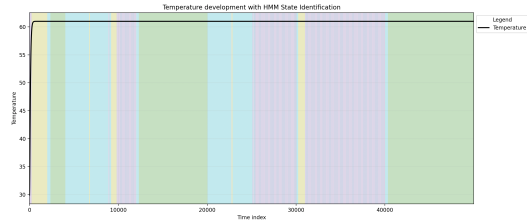
(b) Pump temperature development.

Figure 5.9: States presented in the temperature data | 4 states, 3 mixtures.

In contrast, the feature set $d_m, d_p, \sqrt{T_m^2 + T_p^2}$ with 30,000 training samples exhibited the highest rate of instability. As shown in Figure 5.10, this configuration resulted in frequent, rapid state transitions throughout the observed period.



(a) Motor temperature development.



(b) Pump temperature development.

Figure 5.10: States presented in the temperature | 7 states, 2 mixtures.

The real time operational state identification for the full and marginalized model is presented in 5.9.

Table 5.9: Real-Time operating state identification performance of the models across different feature sets (50,000 samples).

Feature Set	Marg. Model Real-Time State Ident.	Full Model Real-Time State Ident.
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}, \sqrt{\Delta T_m^2 + \Delta T_p^2}\}$	98.87%	99.73%
$\{S_m^2, S_p^2, \sqrt{T_m^2 + T_p^2}, \sqrt{\Delta T_m^2 + \Delta T_p^2}\}$	32.71%	58.37%
$\{S_m^2, S_p^2, \sqrt{T_m^2 + T_p^2}\}$	82.93%	99.08%
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$	26.10%	28.35%

in summary, the results presented shows how different feature sets affects the result on simulated data.

Empirical Data Comparison

The model's performance on empirical data is presented in Table 5.10, highlighting its capability to maintain state consistency after marginalization. The results demonstrate that all evaluated feature sets retain a significant portion of the system information, with match ratios ranging from 90.59% to 99.73%.

Table 5.10: HMM-GMM results for real data. Match ratio indicates the consistency between the full and marginalized model on the same measurement session.

Feature Set	States/Mixtures	Failures	Match Ratio
$\{d_m, d_p, T_m^2, T_p^2\}$	3/1	16/18	94.35%
$\{S_m^2, S_p^2, \sqrt{T_m^2 + T_p^2}\}$	7/1	9/18	90.59%
$\{S_m^2, S_p^2, T_m, T_p\}$	3/1	16/18	94.09%
$\{d_m, d_p, T_m, T_p\}$	2/1	17/18	99.73%
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$	4/2	9/18	95.97%
$\{S_m, S_p, T_m, T_p\}$	3/1	17/18	93.01%

The real time operational state identification for the full and marginalized model on empirical data, presented in Table 5.11, shows that similar to simulated data the marginalized and full model can identify operational states in real time, when preformed between measurement session.

Table 5.11: Real-Time operating state identification performance for empirical data across different feature sets.

Feature Set	Marg. Model Real-Time State Ident.	Full Model Real-Time State Ident.
$\{d_m, d_p, T_m^2, T_p^2\}$	42.56%	61.82%
$\{S_m^2, S_p^2, \sqrt{T_m^2 + T_p^2}\}$	78.37%	78.37%
$\{S_m^2, S_p^2, T_m, T_p\}$	32.43%	67.23%
$\{d_m, d_p, T_m, T_p\}$	38.77%	38.77%
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$	49.66%	7.09%

The section shows that different feature sets provides different results regarding state identification in real time on empiric data.

5.2.4 Anomaly Detection

In this section, we present the performance of HMM-GMMs in detecting anomalies. We evaluate the models' effectiveness using a combination of simulated and real data.

Simulated Data

To demonstrate the effectiveness of the anomaly detection, new data was simulated with lower levels of temperature deviation, as shown in Figure 5.11. The initial 2700 time indices were used for training, and abnormal behavior was introduced at index 3000. The resulting anomaly counts for this setup are presented in Table 5.12. Additionally, other feature sets also scored high anomaly identification rates, ranging from 77.0% to 99.9%.

5. Results

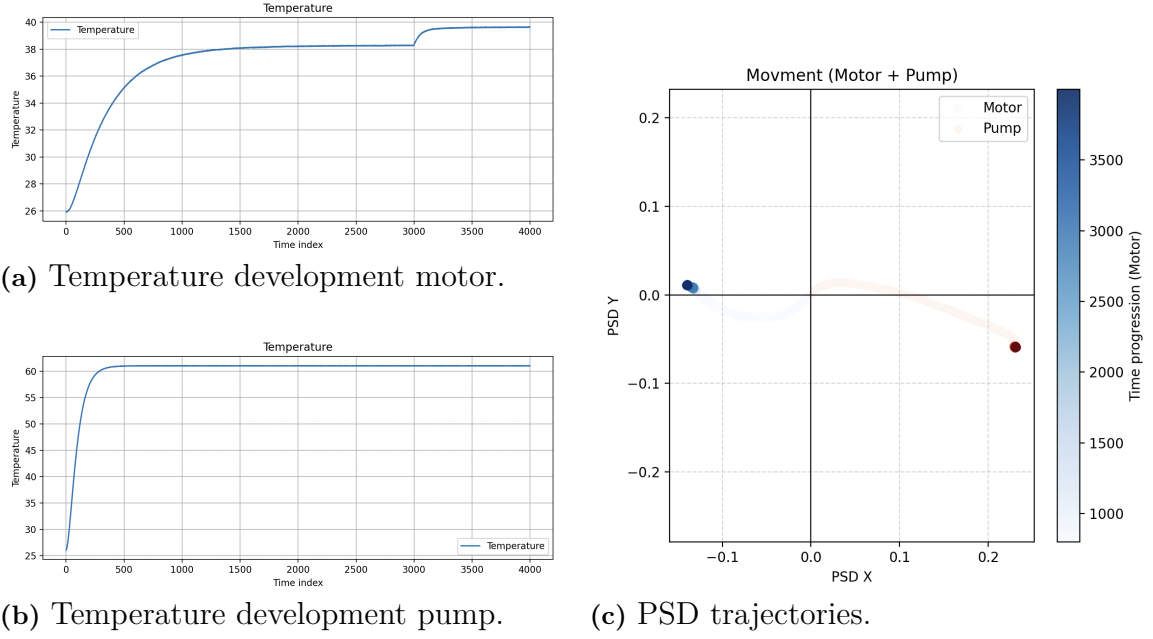


Figure 5.11: Simulated results showing temperature and PSD behavior of simulation with added abnormal behavior.

Table 5.12: Anomaly detection on simulated data. The baseline feature set is used.

Feature Set	Marg. Mod. Anomalies	Full. Mod. Anomalies
$\{d_m, d_p, \sqrt{T_m^2 + T_p^2}, \sqrt{\Delta T_m^2 + \Delta T_p^2}\}$	999	845

Several other feature sets achieved similar detection accuracies between 77.0% and 99.9% across both models.

Empirical Data

Unlike the simulated case, the empirical datasets do not contain intentional faults. Instead, anomaly detection is used here to evaluate the models' sensitivity to distributional shifts between two different measuring sessions.

By considering two measurements sequences that have been sampled at different measuring sessions, $\mathcal{D}_{\text{real}_A}$ and $\mathcal{D}_{\text{real}_B}$, presented in Figure 5.12. Since the baseline feature set does not provide any models with numerical stability, the most promising feature set related to anomaly detection in empirical data is presented, $\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$.

As shown in Table 5.13, when training is done on $\mathcal{D}_{\text{real}_A}$, both the full and marginalized model identifies significant amounts of anomalies. However, the reverse test yielded 0 anomalies for the marginalized model and 2 for the full model. It shows that the marginalized model is less sensitive to deviations than the full model.

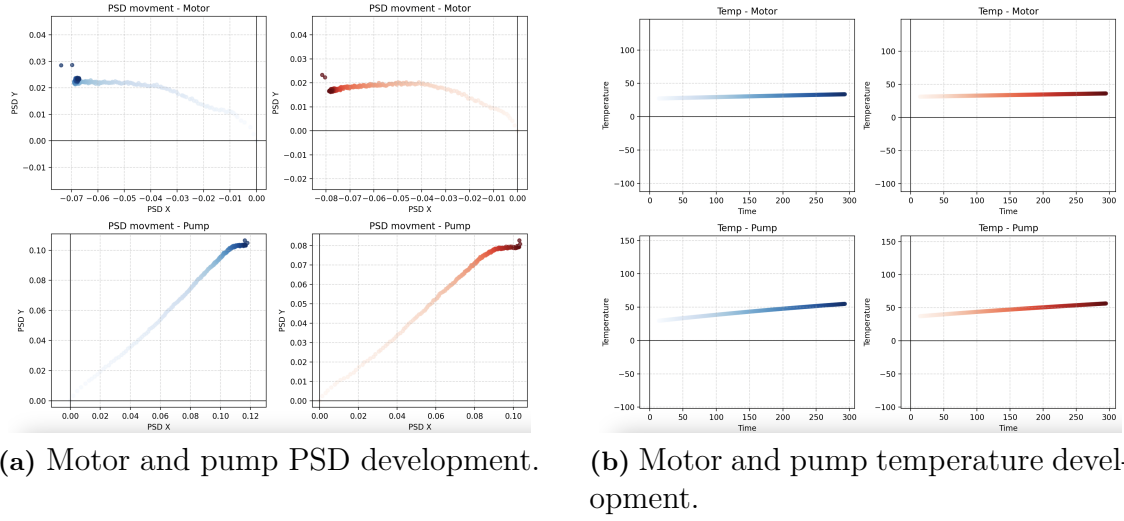


Figure 5.12: Data from two different measuring sessions | (Blue: $\mathcal{D}_{\text{real}_A}$, Red: $\mathcal{D}_{\text{real}_B}$).

Table 5.13: HMM-GMM results for anomaly detection on empirical data. The feature set $\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$ is used.

Training Dataset	Predict Dataset	Marg. Mod. Anomalies	Full. Mod. Anomalies
$\mathcal{D}_{\text{real}_A}$	$\mathcal{D}_{\text{real}_B}$	44	267
$\mathcal{D}_{\text{real}_B}$	$\mathcal{D}_{\text{real}_A}$	0	2

Other configurations resulted in the full models classifying nearly all data points as anomalies, as can be seen in Table 5.14, where an additional set was evaluated.

Table 5.14: HMM-GMM results for anomaly detection on empirical data. The feature set $\{S_m, S_p, \sqrt{T_m^2 + T_p^2}\}$ is used.

Training Dataset	Predict Dataset	Marg. Mod. Anomalies	Full. Mod. Anomalies
$\mathcal{D}_{\text{real}_A}$	$\mathcal{D}_{\text{real}_B}$	35	267
$\mathcal{D}_{\text{real}_B}$	$\mathcal{D}_{\text{real}_A}$	49	265

5.3 Pipeline Evaluation

The models utilized in the pipeline evaluation consist of the tuned LSTM model, presented in Table 5.5 and 5.8, and an HMM-GMM model trained on the feature set $\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$. This specific feature set was selected as it provides a stable model for both real and simulated data, as well as using necessary features (d_m, d_p) to use the predictions from the LSTM

Figure 5.13 depicts the results of the full pipeline applied to an unseen simulation of 2880 samples with known ground-truth behavior. At $t=1900$, synthetic anomalies

were introduced via a step increase in RPM, which subsequently induced a rise in temperature and a divergence in the PSD magnitude from the baseline trace. The detection performance of the full HMM-GMM architecture is illustrated in Figure 5.14, while Figure 5.15 presents the results for the marginalized HMM-GMM variant.

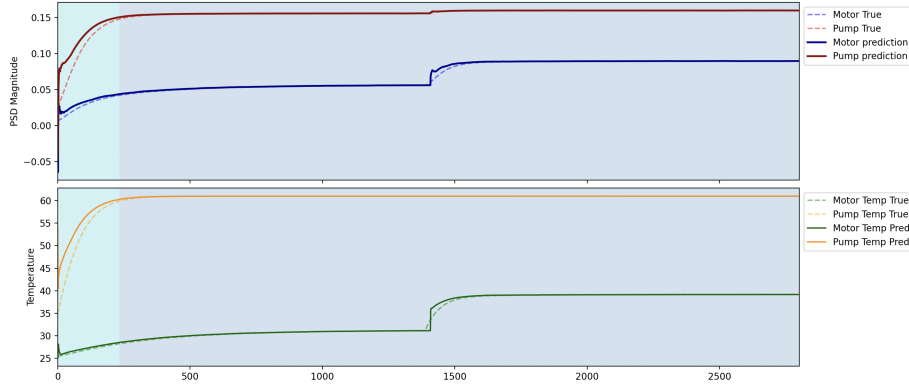


Figure 5.13: Pipeline evaluation on simulated data. The upper plot displays PSD magnitudes for motor and pump, while the lower plot shows temperature development. Background colors indicate the HMM-GMM identified states, where during training 2 states and 1 mixture was identified. Without anomaly inclusion.

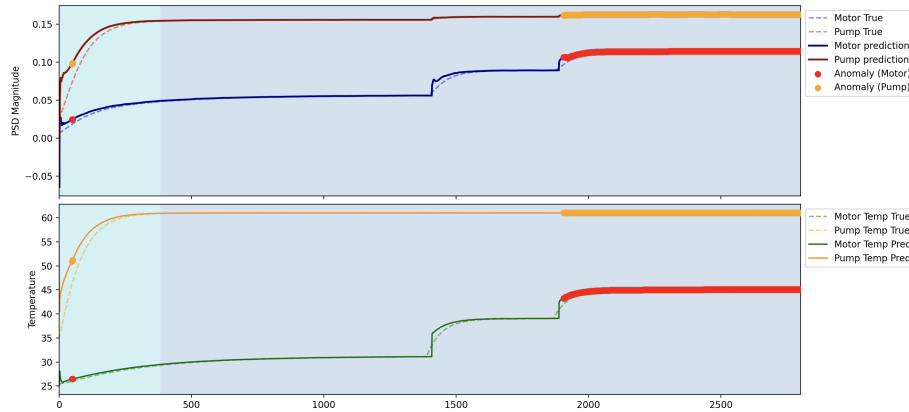


Figure 5.14: Full pipeline using full HMM-GMM model, with anomalies introduced.

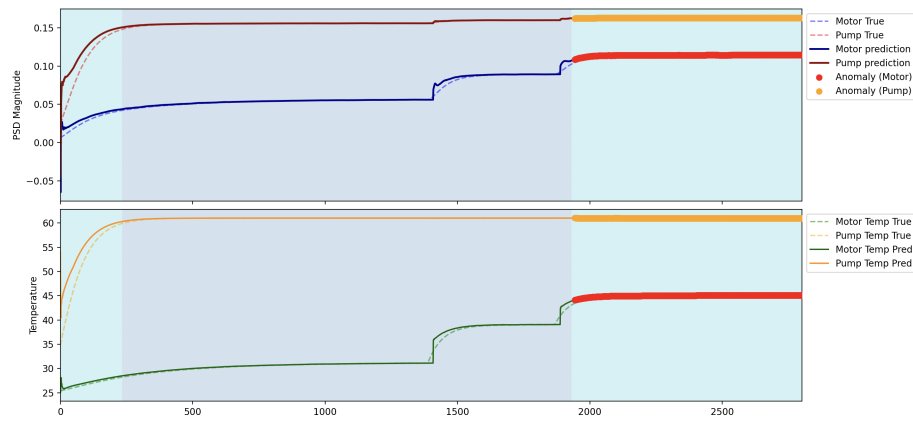


Figure 5.15: Full pipeline using marginalized HMM-GMM model, with anomalies introduced.

Table 5.15: Pipeline results for anomaly detection for full and maginalized models.

	Anomalies Detected	Real-time State Ident.
Full HMM-GMM / LSTM	947/1000	96.61%
Marg. HMM-GMM /LSTM	907/1000	60.19%

Both models identified the abnormal behavior, with the full HMM-GMM detecting the deviation with a shorter temporal latency compared to the marginalized approach, as shown in Figure 5.14 and 5.15. Furthermore, the full model achieved a real-time state identification accuracy of 96.61%, while the marginalized model reached 60.19%, presented in Table 5.15

The results from running the pipeline on empirical data are presented in Figure 5.16.

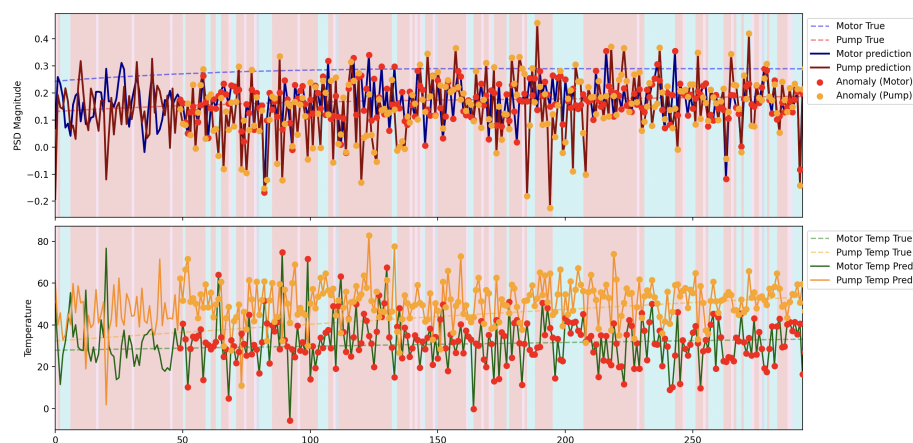


Figure 5.16: Pipeline evaluation on empirical data. Background colors indicate the HMM-identified states, where during training 4 states and 3 mixture was identified.

The HMM-GMM identifies 293/341 samples as abnormal. While the state identification real time match ratio drops to 27.99%, this is largely attributed to the

presence of severe high-frequency state switching.

In conclusion, the results demonstrate the individual performance of the LSTM model for predictions and the HMM-GMM for anomaly detection and state identification, as well as the operational synergy achieved through the combined pipeline. Having presented these findings, the subsequent chapter, *Discussion*, analyzes the implications of these results, compares the strengths and limitations of each approach, and addresses sources of error.

6

Discussion

In this thesis, a pipeline has been developed and evaluated for identification of operating conditions and misalignment. A HMM-GMM model was created for state identification and anomaly detection in the pipeline and an LSTM was created for predictions. The HMM-GMM was designed to handle data in the form of only temperature readings with or without PSD values.

The discussion will cover the implications of the results, evaluate the proposed algorithm and look into future works.

6.1 LSTM as a Method for Prediction of Alignment

The LSTM architecture shows promising performance for prediction of misalignment, given the simulated model developed, as it is capable of tracing the actual path of the simulation with an overall low MAE of $\approx 0.01 - 0.02$ [mm]. Though noting that the LSTM is trained on a simulated model multiple sources can be identified which causes the performance to diverge when prediction of empiric data, as seen by the predictions illustrated in Figure 5.16.

One of the major cause for the poor prediction on empiric data is due to model mismatch, caused by complex relations in the physical system not being fully captured in the simulated model. As mentioned in Section 2.3 the data collected was constrained to periods of 30 minutes, and the simulation assumes sequences collected during longer periods, dynamics modeled in the simulation does not capture all of the dynamics of the real system.

Another thing of note is that when the RPM of the machine is changed in the simulation the LSTM is unable to accurately predict the PSD offsets, see example Figure 5.5. This as the LSTM does not have the information about the rotation speed change in advance, as RPM data as it is presented to the LSTM as real time data. The information about RPM changes could possibly be provided to the LSTM in advance to enable enhanced tracking of the system dynamics, through deeper integration with the control system of the machinery.

One notable characteristic of the models trained with MSE is that the model assumes that there exists a linear dependency between the temperatures T_m and T_p .

This assumed dependency becomes clear after the RPM change occurs, see models utilizing MSE in Figure 5.3. It is possible to see a steep increase in temperature in the pump for models trained with MSE followed by a decline over time, whilst the actual temperature of the pump is constant. This decline in temperature is similar to the temperature decline of the motor for these models, suggesting some linear relation between the temperatures. This type of relationship might hold during the warm-up and stationary phase prior to the change in RPM, though once the motor RPM changes this no longer holds. One of the reasons for this linear relation appearing in the model is the fact that for most of simulations T_p is close to constant, leading to gradients close to zero. Limiting the learning capacity of the LSTM.

The simulation results demonstrate that the past PSD offsets (d_m, d_p) are the most influential features for predictive performance connected to the PSD estimations. These offsets provide a reference for the current PSD magnitude, effectively bounding the estimation. This dependency is highlighted by the lacking performance from feature case 3 and 4, which are not able to track offsets in PSD's accurately. It should be noted, however, the simulation contains high variance between each generated sequence, due to multiple parameters being sampled from uniform distributions, practically representing loads of different machines. For example, looking at the parameters connected to thermal expansion $(\alpha_m, \beta_m, \alpha_p, \beta_p)$ may impose a significant variation in the magnitude of which the PSD's vary. Narrowing these down to even smaller ranges may thus more accurately represent how one physical machine moves. This assumes that the expansion parameters do not change over time, ensuring that the machinery's movement retain a higher similarity between running cycles. A more similar set of sequences to train hypothetically should lead to increased performance for the models which does not utilize the PSD offsets as input variable.

6.2 HMM-GMM as a Method for State Identification and Anomaly Detection.

The results highlight a significant discrepancy between simulated and empirical data regarding feature performance. In simulations, temperature derivatives were superior, likely because they added dimensions that improved state separation. However, these same features caused numerical instability in real-world applications due to the linear nature of the temperature growth and limited data volume. For real-world state identification, the feature set $\{S_m, S_p, \sqrt{T_m^2 + T_p^2}\}$ emerged as the most robust.

In contrast, the set $\{d_m, d_p, \sqrt{T_m^2 + T_p^2}\}$ demonstrated the highest consistency across different measuring sessions. By avoiding raw temperature values and instead using squared or combined features, the model achieved better data separation and increased variance, which was necessary to avoid singular covariance matrices. Nevertheless, no single feature set provided optimal results across all test scenarios, suggesting that feature selection must be tailored to the specific operational context.

Beyond the choice of features, the choice of training data significantly impacts the

model’s ability to generalize. The inconsistency observed between empirical measuring sessions suggests that the machinery’s behavior during the startup phase is highly variable. Since the datasets were acquired during startup, a period characterized by fluctuating alignment and temperature, they likely captured distinct operating modes that were not present in every session. This explains why models trained on one session often perceived the normal operational variance of another session as anomalous. For determining the performance of the proposed method in real-world scenarios, more data and longer measuring sessions is needed

Despite these challenges in generalization, the marginalized model demonstrated a strong capability to preserve the underlying state structure of the full model, with state reconstruction ratios often exceeding 90% (Table 5.8). This indicates that the HMM-GMM framework is effective at maintaining diagnostic integrity even when sensor dimensionality is reduced. However, real-time state identification remains highly dependent on the full system context. As shown in Tables 5.9 and 5.11, the full model consistently outperformed the marginalized version. The wide range in real-time match ratios (from 7.09% to 99.73%) underscores the sensitivity of the models.

Finally, the tendency of the Bayesian Information Criterion (BIC) to favor an excessive number of states and mixtures points toward a risk of overfitting. Significant fluctuations in a single feature can lower the likelihood, causing the model to fragment cohesive behavior into several substates. This sub-optimal segmentation suggests that while the HMM-GMM is powerful, it requires careful regularization or more extensive datasets to ensure that the identified states correspond to physical realities rather than numerical noise.

6.3 Comparative Analysis of Pipeline Performance

The proposed pipeline yields fairly accurate results for the simulated data, as illustrated in Figure 5.3. By integrating both PSD measurements and temperature (see Figure 5.14), the algorithm identifies deviations earlier than when relying exclusively on temperature data (see Figure 5.15). This improvement is likely due to the additional system information provided by the full HMM-GMM model compared to the marginalized version, which also enables the detection of two initial anomalies triggered by deviated behavior of the initial LSTM predictions. The results also demonstrate that the full model in general achieves higher accuracy in real-time operating state identification, as seen in 5.9. However, when comparing these results to the pipeline’s performance on empirical data (see Figure 5.16), certain discrepancies are observed. As previously addressed in Section 6.1, the simulated data does not fully capture the complexities of the real-world environment. This limitation affects the LSTM’s ability to accurately predict temperature and alignment, which in turn leads to less reliable state predictions and the identification of abnormal behavior throughout the entire sequence.

Although the evaluation was illustrated with one single simulated and real dataset,

the overall performance was highly correlated to the individual performances of the different modules. Thus refinement on the individual modules and simulation model, is assumed to have a positive effect on the pipelines performance.

6.4 Research Questions

The following discussion evaluates the simulation results based on the research questions defined in Section 1.3.

Research Question 1

The results demonstrate that predicting alignment magnitude is feasible using the developed simulation based on an LSTM architecture. However, direct prediction of empirical data was not possible due to a model mismatch. If a significant increase in empirically collected data were achieved, enough to train the LSTM. The LSTM architecture is expected to be suitable for the task, provided that the features described in Case 5 are included. This implies that laser measuring devices must remain active for accurate real-time forecasting.

The model achieved an accuracy error within the 0.01–0.02 mm range for simulated data, applying this to empirical scenarios assumes that machinery dynamics remain stationary over time, LSTMs predictions are inherently limited to patterns encountered during training. Furthermore, the current model does not account for individual PSD components (x,y). Consequently, specific machinery movement cannot be predicted without further architectural development.

Research Question 2

The results demonstrate that operating states can be identified to an extent using both independent temperature readings and combinations with laser measurements. Specifically, the simulated data showed that real-time state identification reached accuracies of up to 99.73% when using the full model context and 98.87% when using the marginalized model context. The evaluation on real-world data further reinforces these findings, where the highest state identification accuracy was 78.37% for both the marginalized and full model. However, even if a high state recognitions accuracy seems plausible for empirical data, models created from all but one feature set identifies the new data as abnormal, as discussed in Section 6.2. Thus, limiting the proposed method in real-world scenarios.

While the models proved capable of detecting abnormal behavior in simulated data even without laser measurements, their performance is notably enhanced by the full system context. While the fundamental capacity for identification of abnormal behavior is established, industrial-grade precision requires further optimization of feature sets and anomaly thresholds on more representative datasets.

6.5 Future Work

For future work the suggestion is to conduct extensive measurement on real machinery to validate the simulated findings in this report. With an increase in measured parameters as complement to the ones used for this study, in order to get a more holistic view of the machine behavior. As well as collecting sufficient amount of data in order to train models on empirically collected data alone is open for further evaluation.

Additional measured parameters that future work could involve, monitoring of the motor's feeding current to capture load fluctuations on the axis, flow and pressure measurements from the pump, and possibly internal temperature readings from bearings, to remove the thermal lag induced by measuring temperature on the chassis surface.

Future work could also involve the measurements of vibrations, as it has proven to provide important information of machine health in the field of PdM, as shown in [1]. Monitoring specific frequency spectra may provide signs of misalignment and mechanical wear, enhancing the knowledge of the machinery.

References

- [1] M. Romanssini, P. C. C. de Aguirre, L. Compassi-Severo, and A. G. Girardi, “A review on vibration monitoring techniques for predictive maintenance of rotating machinery,” *Eng*, vol. 4, no. 3, pp. 1797–1817, 2023. DOI: 10.3390/eng4030102. [Online]. Available: <https://www.mdpi.com/2673-4117/4/3/102>.
- [2] H. Habbouche, T. Benkedjouh, Y. Amirat, and M. Benbouzid, “A wavelet transform-based transfer learning approach for enhanced shaft misalignment diagnosis in rotating machinery,” *Electronics*, vol. 14, no. 2, p. 341, 2025. DOI: 10.3390/electronics14020341. [Online]. Available: <https://www.mdpi.com/2079-9292/14/2/341>.
- [3] Easy-Laser AB. “Product overview: XT770 shaft and geo alignment system.” Retrieved January 20, 2026, Easy-Laser AB. [Online]. Available: <https://easylaser.com/en-us/products/shaft-alignment/xt770-shaft-geo?t=downloads>.
- [4] Z. Liu, E. Blasch, M. Liao, C. Yang, K. Tsukada, and N. Meyendorf, “Digital twin for predictive maintenance,” Apr. 2023, p. 6. DOI: 10.1117/12.2660270.
- [5] D. Zhong, Z. Xia, Y. Zhu, and J. Duan, “Overview of predictive maintenance based on digital twin technology,” *Helvion*, vol. 9, no. 4, e14534, 2023. DOI: 10.1016/j.helivon.2023.e14534. [Online]. Available: <https://doi.org/10.1016/j.helivon.2023.e14534>.
- [6] O. Oladele and J. Gilbert, “Ethical considerations in data-driven product management: Balancing innovation and responsibility,” Oct. 2023.
- [7] H. Ma and M. Su, “Whom to sue? liability of unaccountability in ai decisions,” *Organizational Dynamics*, vol. 54, no. 3, Part 2, p. 101 123, 2025, ISSN: 0090-2616. DOI: <https://doi.org/10.1016/j.orgdyn.2024.101123>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0090261624000986>.
- [8] A. Benhanifia, Z. B. Cheikh, P. M. Oliveira, A. Valente, and J. Lima, “Systematic review of predictive maintenance practices in the manufacturing sector,” *Intelligent Systems with Applications*, vol. 26, p. 200 501, 2025, ISSN: 2667-3053. DOI: <https://doi.org/10.1016/j.iswa.2025.200501>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2667305325000274>.
- [9] M. Abu-Zeid and S. Abdel-Rahman, “Bearing problems’ effects on the dynamic performance of pumping stations,” *Alexandria Engineering Journal*, vol. 52, pp. 241–248, Sep. 2013. DOI: 10.1016/j.aej.2013.02.002.

- [10] O. Tonks and Q. Wang, “The detection of wind turbine shaft misalignment using temperature monitoring,” *CIRP Journal of Manufacturing Science and Technology*, vol. 17, pp. 71–79, 2017, ISSN: 1755-5817. DOI: 10.1016/j.cirpj.2016.05.001.
- [11] M. Liggins, D. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice* (The Electrical Engineering and Applied Signal Processing Series), 2nd. Boca Raton, FL: CRC Press, 2009, ISBN: 9781420053081.
- [12] E. Wibowo, N. Ulya, M. Farizi, and N. Fitriyanti, “Derivation of newton’s law of cooling and heating: Heating the water then cooling it down naturally to the room temperature,” *Momentum: Physics Education Journal*, vol. 7, Jan. 2023. DOI: 10.21067/mpej.v7i1.6889.
- [13] D. Febriansyah, B. Tambunan, and N. Budi, “Influence of thermal expansion on steam turbine shaft alignment (influence of thermal expansion on steam turbine shaft alignment pengaruh pemuaian panas terhadap kelurusan poros turbin uap,” Apr. 2020. DOI: 10.29122/mipi.v14i1.3971.g3416.
- [14] M. H. Albderi and I. Szabo, “Experimental study of the effect of misalignment on rolling element bearing,” *European Journal of Engineering and Technology Research*, vol. 6, no. 7, pp. 87–90, 2021. DOI: 10.24018/ejeng.2021.6.7.2661.
- [15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.
- [16] H. Sak, A. Senior, and F. Beaufays, *Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition*, 2014. arXiv: 1402.1128 [cs.NE]. [Online]. Available: <https://arxiv.org/abs/1402.1128>.
- [17] F. Deloche, *A diagram for a one-unit Long Short-Term Memory (LSTM)*, Wikimedia Commons, Accessed: 31 Mar. 2026, 2017. [Online]. Available: https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg.
- [18] A. Paszke et al., “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.
- [19] T. N. Truong et al., “Why globally re-shuffle? revisiting data shuffling in large scale deep learning,” May 2022, pp. 1085–1096. DOI: 10.1109/IPDPS53621.2022.00109.
- [20] A. Pranolo et al., “Enhanced multivariate time series analysis using lstm: A comparative study of min-max and z-score normalization techniques,” *ILKOM Jurnal Ilmiah*, vol. 16, pp. 210–220, Aug. 2024. DOI: 10.33096/ilkom.v16i2.2333.210-220.
- [21] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/1412.6980>.
- [22] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY: Springer, 2006, ISBN: 978-0-387-31073-2.
- [23] A. Jadon, A. Patil, and S. Jadon, *A comprehensive survey of regression based loss functions for time series forecasting*, 2022. arXiv: 2211.02989 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2211.02989>.

-
- [24] A. Tavares et al., “Online distribution-based clustering algorithm for condition monitoring in rotating machines,” *IEEE Access*, 2025, IEEE Xplore: <https://ieeexplore.ieee.org/document/11397595>.
 - [25] J. Yu, “Health condition monitoring of machines based on hidden markov model and contribution analysis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 8, pp. 2200–2211, 2012. DOI: 10.1109/TIM.2012.2184015.
 - [26] X. Qiu, W. Chen, Q. Wu, Y.-W. Wang, C. Gu, and W.-A. Zhang, “Fault diagnosis for multi-axis carving machine systems with gaussian mixture hidden markov models: A data-model interactive perspective,” *Control Engineering Practice*, vol. 154, p. 106 163, 2025, ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2024.106163.
 - [27] H. Mostafaei and M. Ghamami, “State of the art in automated operational modal identification: Algorithms, applications, and future perspectives,” *Machines*, vol. 13, no. 1, p. 39, 2025. DOI: 10.3390/machines13010039.
 - [28] J. A. Bilmes, “A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models,” International Computer Science Institute, Berkeley, CA, Tech. Rep., 1998, Available: http://leap.ee.iisc.ac.in/sriram/teaching/MLSP19/refs/GMM_Bilmes.pdf.
 - [29] D. Cournapeau, F. Pedregosa, G. Varoquaux, S. Lebedev, A. Lee, M. Danielson, et al., *Hmmlearn: Hidden markov models in python, with scikit-learn like api*, version 0.3.3, Accessed: 2026-04-24, 2026. [Online]. Available: <https://github.com/hmmlearn/hmmlearn>.
 - [30] J.-G. Lee, D.-H. Kim, and J. H. Lee, “Proactive fault diagnosis of a radiator: A combination of gaussian mixture model and lstm autoencoder,” *Sensors*, vol. 23, no. 21, p. 8688, 2023. DOI: 10.3390/s23218688.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY